

NPS69-86-005

NAVAL POSTGRADUATE SCHOOL

Monterey, California



CONTRACTOR REPORT

THRUST VECTOR CONTROL,

HEAT TRANSFER MODELING

by

A. Leitner

July, 1986

Approved for public release; distribution unlimited.

Prepared for:
Naval Postgraduate School
Monterey, California 93943

FEDDOCS
D 208.14/2:
NPS-69-86-005

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

NAVAL POSTGRADUATE SCHOOL
Monterey, California

RADM R. C. Austin
Superintendent

D. A. Schradly
Provost

The work reported herein was carried out for the Naval Postgraduate School by Mr. Amiram Leitner under Contract Number N62271-86-M-0206. The work presented in this report is part of a project sponsored by Naval Weapons Center on heat transfer modeling and is under the cognizance of Professors Matthew D. Kelleher and Robert H. Nunn.

Reproduction of all or part of this report is authorized.

This report was prepared by:

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS	
a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
b DECLASSIFICATION/DOWNGRADING SCHEDULE		5 MONITORING ORGANIZATION REPORT NUMBER(S)	
PERFORMING ORGANIZATION REPORT NUMBER(S) NPS69-86-005		7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable)	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	
c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N62271-86-M-0206	
a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Weapons Center	8b OFFICE SYMBOL (If applicable)	10 SOURCE OF FUNDING NUMBERS	
c ADDRESS (City, State, and ZIP Code) China Lake, California 93555		PROGRAM ELEMENT NO	PROJECT NO
		TASK NO	WORK UNIT ACCESSION NO
1 TITLE (Include Security Classification) Thrust Vector Control, Heat Transfer Modeling			
2 PERSONAL AUTHOR(S) A. Leitner			
3a TYPE OF REPORT	13b TIME COVERED FROM TO	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT 65
6 SUPPLEMENTARY NOTATION			
COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
9 ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>The report presents heat transfer modeling of Thrust Vector control systems using the PHOENICS computer code.</p> <p>Simple two-dimensional wedge and blunt bodies have been examined in supersonic cold flow, for both laminar and turbulent flow cases.</p> <p>The research presents a numerical solution of the supersonic compressible viscous two-dimensional flow field. Post calculations were done to estimate skin friction coefficient, surface heat flux, heat transfer coefficient and Stanton number distributions in both wedge and blunt cases.</p>			
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL		22b TELEPHONE (Include Area Code)	22c OFFICE SYMBOL

Thrust Vector Control Heat Transfer Modeling

Abstract

The report presents heat transfer modeling of Thrust Vector control systems using the PHOENICS computer code.

Simple two-dimensional wedge and blunt bodies have been examined in supersonic cold flow, for both laminar and turbulent flow cases.

The research presents a numerical solution of the supersonic compressible viscous two-dimensional flow field. Post calculations were done to estimate skin friction coefficient, surface heat flux, heat transfer coefficient and Stanton number distributions in both wedge and blunt cases.

NOMENCLATURE

C_p	Specific heat [J/kg·k]
C_1, C_2, C_D	Constants used in turbulent model
C_f	Skin friction coefficient
h	Enthalpy [J/kg]
h_c	Heat Transfer coefficient [W/m ² k]
M	Mach number
P	Pressure
Pr	Prandtl number
q	Heat flux
R	Gas constant [J/kg·k]
Re	Reynolds number
St	Stanton number
t	Time [S]
T	Temperature

GREEK LETTER SYMBOLS

γ	Specific heat ratio
δ	Boundary layer thickness
μ	Dynamic viscosity [kg·m/s]
σ	General exchange coefficient
ρ	Density [kg/m ³]
τ_k, τ_ϵ	Constants used in turbulent model
ϕ	Any property at the grid node

SUBSCRIPTS

comp	Compressible value
eff	Effective value
inc	Incompressible value

r	Recovery
lam	Laminar quantity
t	Turbulent quantity
stat	Static values
W	Wall value
z	Local value in the flow direction
∞	Free stream value

Table of Contents

1. Introduction	1
2. PHOENICS description	4
2.1 The structural principle of PHOENICS	5
2.2 Numerical scheme	5
3. Geometry and Dimensions	6
4. Assumptions	7
5. Governing equations	10
6. Input Variables	12
7. Boundary conditions	14
7.1 Inlet	14
7.2 Outlet	14
7.3 Free stream boundary	15
7.4 Solid wall	15
7.5 Wall function	15
7.6 Boundary conditions in PHOENICS	16
8. Mesh generation	18
9. Heat Transfer Analysis	20
10. Code and Computer	23
11. Results discussion	24
12. Conclusions and Recommendations	25
List of References	26
Appendices	
A - Satellite listing	38
B - Ground listing	57
C - Digital results	

List of Figures

1. NWC jet vane configuration	27
2. Jet vane approximation	28
3. Wedge vane domain and grid	29
4. Blunt vane domain and grid	30
5. Re_x No. along the vane	31
6. C_f in laminar flow	32
7. C_f in turbulent flow	33
8. S_t in laminar flow	34
9. S_t in turbulent flow	35
10. Coefficient of heat convection in laminar flow	36
11. Coefficient of heat convection in turbulent flow	37

1. Introduction

This report describes a numerical analysis of heat transfer of a typical jet vane configuration used for thrust vector control. The work was carried out under contract Nos. N62271-85-M-0443 and N62271-86-M-0206, for the Naval Postgraduate School.

The tasks to be accomplished under the first contract were:

Task I: Formulate the conservation equations of momentum energy for two-dimensional, supersonic flow in geometries typical of thrust vector control systems.

Task II: Formulate boundary conditions for these equations appropriate to thrust vector control systems.

The tasks to be accomplished under the second contract were:

Task I: Continue and update the formulation of thrust vector control geometries based on the input from the Naval Weapons Center (NWC).

Task II: Construct the computational model for implementation in the PHOENICS code, of the thrust vector control geometries and flow conditions provided by NWC.

Task III: Run the PHOENICS code for the previously formulated models. Analyze and interpret the PHOENICS results for surface temperature and heat flux.

Thrust vector control components such as jet vanes and jet tabs are exposed to high speed hot gases at the exit of a rocket nozzle.

Estimation of the heat transfer from the hot exhaust gases to the vane is major consideration in the correct design of a vane, and its ability to survive during its mission.

The research work was done under the framework of the tasks. A brief survey of what has been done according to the task is given:

Task I (M-0443): Heat transfer modeling of thrust vector control vane requires supersonic compressible viscous flow analysis.

In order to meet the requirements, the conservation differential equations of mass momentum energy and the two $k-\epsilon$ turbulent equations were formulated, and additional algebraic formulas for the relations between pressure density and the equation of state for ideal gas.

Task II (M-0443): The physical dimensions of the flow field grid were chosen and the boundary conditions for the Navier-Stokes, energy and the two $k-q$ turbulent model equations were given.

Task I (M-0206): Working on the task, the actual configuration of a jet vane that is presently being tested at NWC has been modeled. The geometry being used is a wedge which has the same half angle and dimensions as the NWC jet vane.

Task II (M-0206): BFC (Body fitted coordinate) version of PHOENICS code (Ref. 3) was used for calculating the flow-field and heat transfer over the model. Using the BFC, a better geometrical approximation to vane shape could be achieved.

Non-Uniform grids have been utilized in order to model complicated regions in the flow field. Relaxation parameters and false timesteps options were adjusted to enable efficient computer runs with good convergence.

Task III (M-0206): In carrying out this task, four major runs have been analyzed:

Two geometric configurations were used: wedge vane and blunt vane (see Figures 1, 2, 3, 4); each one in both laminar and turbulent flow conditions.

Numerical results for fluid field and thermodynamic properties of pressure, temperature, density, Mach number and velocities are given in appendix C.

Post-calculations of heat transfer coefficient, skin friction coefficient and Stanton number are given in Figures (6, 7, 8, 9, 10, 11).

The next chapters describe in more detail the process of building the model and the analysis of the results.

2. PHOENICS Description

The present work addresses the heat transfer modeling of thrust vector control systems. In this effort the Navier-Stokes approach is applied by using a computer code which is capable of simulating a large number of fluid flow, heat transfer and chemical reaction processes which arise in industry and elsewhere. This code is called PHOENICS, which is an acronym standing for: 'Parabolic, Hyperbolic or Elliptic Numerical Integration Code Series.' The name comes from the fact that the differential equations of fluid flow, etc. arise in forms classified by mathematicians as parabolic, hyperbolic or elliptic; and PHOENICS solves these equations, whatever their form.

Built into PHOENICS are the major conservation laws of physics (mass, momentum, and energy) applied to a large number of continuous subdomains called 'cells,' into which the domain of study is artificially divided. The number of cells can be few or many according to the requirements of the problem. Because of numerical stability considerations the restrictions on cell refinement can become particularly burdensome in the calculation of a turbulent boundary layer where a very fine mesh near the wall may be required.

When supplied with appropriate information concerning: the physical properties of the materials, the geometrical and other constraints, the inlet and/or initial conditions, PHOENICS computes the corresponding solutions to the relevant differential equations, expressing them as tables of numbers describing the field of velocity, temperature concentration, etc.

Detailed information about PHOENICS is given in [Ref. 3].

2.1 The Structural Principle of PHOENICS

The code consists of three major parts: Satellite subroutine, Ground subroutine and Earth library.

The satellite subroutine is the main input subroutine and should provide the answers to the questions:

- what kind of process is to be simulated
- what are the properties of the fluid
- what are the shape and size of the domain
- how fine is the grid to be employed
- to what degree of accuracy is the calculation to be continued
- and what output should be provided

Ground subroutine is active during the computing process and is used for updating properties which vary with time, temperature, etc. For example: viscosity depends on temperature or density depends upon pressure and temperatures, etc.

Earth library is the main solver generator. It is given as a binary library and does not enable the user access to the source code.

2.2 Numerical Scheme

The numerical scheme used by the code is the simpler (semi-implicit method for pressure-linked equations revised) (Ref. 9). The scheme was developed by Patankar, S. V. and Spalding, D. B.

The scheme requires an additional dependent variable, the pressure correction, which has no physical meaning but should take part in the process.

The value of the pressure correction should tend to zero in the convergence process.

Two additional differential equations are solved: for the pressure, and for the pressure correction.

3. Geometry and Dimensions

Symmetrical 2-D planar geometry, which is shown in Figure 2, was chosen to be the approximation of the MWC vane in Figure 1.

Two geometrical profiles were examined, one with wedge leading edge and the second with blunt leading edge.

The dimensions of the domain in Figure 3 and 4 satisfy aspect ratio of 10:1 in the vertical y coordinate. A high aspect ratio in the coordinate is important for the assumption of free stream conditions at the upper boundary.

4. Assumptions

Postulating the right or the wrong assumptions has the most influence on modeling process. The stage was carried out very carefully in order to make the most compatible model with reality.

4.1 Steady state:

The modeling assumes steady state physical phenomenon process.

$$\frac{\partial}{\partial t} (\text{all properties}) = 0$$

This is a valid assumption since the time constant for the convection process is much shorter than the time constant for the wall conduction.

By assuming the wall temperature to be constant, the two procedures are decoupled.

In hot flow it is important to run the code for a wide range of wall temperature which will take into account the influence of different temperatures on the heat convection process.

4.2 Cold Air Flow

Ambient temperature air flow which was utilized by NWC experiments is being used in the computations.

4.3 Ideal Gas

The gas is assumed to satisfy the ideal gas equation of state

$$p = \rho RT \tag{4.1}$$

This is a fairly good assumption for nonreactive gas flow. In spite of the values of static temperature can decrease to 200[k], the density remain relatively low.

This assumption is an important simplification to the solution in Ref. 10 which used the isentropic relation between pressure and density instead

$$\frac{\rho}{\rho_0} = \left(\frac{P}{P_0}\right)^{1/\gamma} \quad (4.2)$$

4.4 Constant Pr, γ :

Prantdl number and γ (ratio of specific heats) were found to have negligible variations in the temperature range of the model. (200k ÷ 350k)

4.5 Varying Viscosity and Thermal Conductivity:

μ and k are much more dependent on temperature especially very close to the solid wall where values of μ and k influence strongly the shear and heat transfer mechanism. To account for the temperature dependence power law relations have been formulated for μ and k .

$$\mu = \mu_0 \left(\frac{T}{T_0}\right)^{0.666} \quad (4.3)$$

$$k = k_0 \left(\frac{T}{T_0}\right)^{0.666} \quad (4.4)$$

4.6 Parallel Flow

Gas flow at the exit of the exhaust nozzle is more likely to be a conic source flow than parallel flow.

If the half angle of the nozzle is small, ($\alpha < 15^\circ$), parallel flow is a good assumption

4.7 Negligible Radiation

Assessments that were done showed that heat convection is at least one order of magnitude greater than heat flux by radiation.

4.8 Laminar and Turbulent Solutions

In order to overcome lack of ability to predict transition, separated laminar and turbulent calculations were done for each case. The turbulent solution utilizes the $(k-\epsilon)$ eddy viscosity model Ref. 5.

4.9 Constant Wall Temperature

The vane wall is assumed to have constant temperature during the time of calculation.

5. Governing Equations

The conservation equations for the compressible flow of the mathematical model consists of a viscous, Newtonian perfect gas consisting of the following six differential equations:

Conservation of Mass:

$$\frac{\partial}{\partial t}(\rho) + \nabla \cdot (\rho \vec{V}) = 0 \quad (5.1)$$

Conservation of momentum:

$$\frac{\partial}{\partial t}(\rho \phi) + \nabla \cdot (\rho \vec{V} \phi - \mu \nabla \phi) = \nabla P \quad (5.2)$$

where ϕ is V or W velocity component for y and z direction.

Conservation of Energy

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \vec{V} h - \frac{\mu}{Pr} \nabla h) = \frac{Dp}{Dt} \quad (5.3)$$

where h is the total enthalpy.

$$h = C_p T_o$$

where T_o is the total temperature

$$T_o = T_{stat} * (1 + \frac{\gamma-1}{2} M^2)$$

In the case of laminar flow the governing equations (5.1), (5.2), (5.3) are sufficient to determine a solution when proper boundary conditions are applied and the equation of state (4.1) is provided.

Turbulence Model:

In turbulent flow it is necessary to hypothesize a turbulence model relating the turbulent viscosity to the other problem variables.

The model used in PHOENICS is the eddy viscosity (k-ε) model [Ref. 3, Ref. 5]. In this model k, the turbulent kinetic energy and ε, the turbulence dissipation rate, are treated as properties of the flow and conservation equations are postulated for these properties. The two conservation equations are: one for k, the kinetic energy of turbulence:

$$\frac{Dk}{Dt} = \frac{\partial}{\partial X_j} \left(\frac{\nu_{eff}}{\sigma_k} \frac{\partial k}{\partial X_j} \right) + G_k - \epsilon \quad (5.4)$$

Second equation for ε, the dissipation rate of turbulence

$$\frac{D\epsilon}{Dt} = \frac{\partial}{\partial X_j} \left(\frac{\nu_{eff}}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial X_j} \right) + \frac{\epsilon}{K} (C_1 G_k - C_2 \epsilon) \quad (5.5)$$

where

$$G_k = \nu_t \left(\frac{\partial \bar{U}_i}{\partial X_j} + \frac{\partial \bar{U}_j}{\partial X_i} \right) \frac{\partial \bar{U}_i}{\partial X_j} \quad (5.6)$$

$$\mu_{eff} = \mu_{lam} + \rho c_\mu k^2 / \epsilon \quad (5.7)$$

c_1 , c_2 , σ_k , σ_ϵ , c_μ are empirical constants which are provided in PHOENICS.

The reason for using the (k-ε) model is because it is the most verified model for engineering applications. It combines simplicity, universality, and realism of predictions in most cases.

Two additional differential equations are solved also in order to satisfy the SIMPLER algorithm as was mentioned in chapter 2.2. The description of the pressure and pressure correction equations is provided by Ref. 9.

6. Input Variables

The properties of mach no. stagnation pressure and temperature of the gas were provided by NWC; additional properties were taken from air tables:

Mach number:

$$M_{\infty} = 3.2$$

Stagnation pressure:

$$P_o = 55.10^5 \text{ [Pa]}$$

Stagnation temperature:

$$T_o = 555.55 \text{ [K]}$$

Gas constant

$$R = 287. \text{ [J/kg}\cdot\text{k]}$$

Specific heat ratio

$$\gamma = 1.35$$

Laminar Prandtl Number

$$Pr = 0.7$$

Turbulent Prandtl Number

$$Pr_t = 0.9$$

Constant Pressure Specific Heat

$$C_p = R/(1-1/\gamma) \text{ [J/kg}\cdot\text{k]}$$

Laminar Viscosity

$$\mu = 0.1716 * 10^{-5} * (T/273.)^{0.666}$$

Thermal Conductivity

$$k = \mu C_p / Pr$$

The gas properties in the inlet boundary are equivalent to the properties at nozzle exit. Inlet properties are calculated from the stagnation values in the combustion chamber. The calculation was done by assuming one dimensional

isentropic expansion from combustion chamber to the nozzle exit (inlet for the vane).

$$\text{Pressure} \quad P_1 = P_0 / \left(1 + \frac{\gamma-1}{2} M^2\right)^{\gamma/\gamma-1} \quad (6.1)$$

$$\text{Temperature} \quad T_1 = T_0 / \left(1 + \frac{\gamma-1}{2} M^2\right) \quad (6.2)$$

$$\text{Density} \quad \rho_1 = P/RT_1 \quad (6.3)$$

$$\text{Enthalpy} \quad h_1 = C_p T_1 \quad (6.4)$$

$$\text{Sonic Velocity} \quad C_1 = \sqrt{\gamma R T_1} \quad (6.5)$$

$$\text{Velocity} \quad W_1 = C \cdot M \quad (6.6)$$

The subscript 1 signifies inlet property

7. Boundary Conditions

The flow field described in Figures 3, 4 has four boundaries, which can be named: inlet, outlet, freestream boundary and solid wall.

Super sonic flows have a hyperbolic mathematical nature. The field consists of influence zones, the flow at every point is governed only by its influence zone, basically by the upwind stream.

As a consequence from the discussion, it's obvious that the outlet boundary condition has no influence on the upstream flow. The boundary values that are given at the outlet are to satisfy some numerical needs only.

7.1 Inlet

Parallel uniform flow with known velocity, enthalpy, pressure, and density: equation (6.1), (6.3), (6.4), (6.6) are given at the left boundary of the grid. In PHOENICS this is specified as the LOW side of the first Z cell.

In turbulent flow, boundary conditions are supplied for k and ϵ . The values that are given are based on empirical values:

$$k_i = 0.0 w_i^2 \quad (7.1)$$

$$\epsilon_i = 0.16 k^{1.5} / (5 * GH) \quad (7.2)$$

where GH is half the vane thickness

7.2 Outlet

As was mentioned previously, the outlet has negligible effect on the results. The only property that is specified at the outlet is the pressure.

7.3 Freestream Boundary

Assuming that the upper boundary is chosen to be far enough away, the default boundary condition option of PHOENICS is used. This implies a line of symmetry where all gradients are zero.

7.4 Solid Wall

Zero velocity and constant wall enthalpy (temperature) are assumed on the wall. In PHOENICS the wall is the SOUTH side of the first y cell. The high enthalpy and velocity gradients near the wall demands a refined grid close to the wall. Values of shear stress and heat flux are calculated to first order accuracy using:

$$\tau_w = \mu \frac{\partial W}{\partial y} \approx \mu \frac{W_1}{\Delta y_1/2} \quad (7.1)$$

$$q_w = \frac{\mu}{Pr} \frac{\partial h}{\partial y} \approx \frac{\mu}{Pr} \cdot \frac{h_1 - h_w}{\Delta y_1/2} \quad (7.2)$$

In turbulent flow, a wall function is used to provide the wall condition for velocity, enthalpy, k, and ϵ

7.5 Wall Function

The wall problem in the numerical computation of flows, especially in turbulent flow, is an old one and most authors have adopted similar techniques. In effect they "bridge over" the region very close to the wall by introducing special functions which are called wall functions. These are often empirical in origin. Accounts may be found in Ref. 11.

The problem arises as follows. Turbulence dies out, close to the wall, because the no slip condition and the rigidity of the wall make all the velocity components fall to zero. The consequence is that the effective

viscosity and other transport properties fall there to their laminar values and the result is a rapid variation with distance from the wall both of the ϕ 's and of their gradients.

Where ϕ signifies general dependent variable, it is possible to compute these variations in detail, by using a computer code such as PHOENICS on two conditions:

(i) the grid points must be packed into the region of steep gradient changes closely enough for sufficient numerical accuracy to be obtained

(ii) the functions appearing in the turbulence model equations must properly represent the influence of local Reynolds number on turbulence.

Under the conditions above, the wall function sequences in the program act as follows:

The Reynolds number is first evaluated, based on the resultant velocity parallel to the wall, on the distance from the wall to the grid node and on density and laminar viscosity. If this Reynolds number is less than 132.25 (the value at which the laminar and turbulent wall function intersect) a laminar wall function is used. If this Reynolds number turns out to be greater than 132.25 the velocity variation is logarithmic and the corresponding shear stress coefficient is evaluated. This corresponds to the commonly used "log law" wall function. [Ref. 4]

7.6 Boundary Conditions in Phoenix

PHOENICS utilizes source terms for creating boundary conditions. The form of the source term of each dependent variable ϕ is:

$$S\phi = ([m] + C\phi) (V\phi - \phi_p) \quad (7.3)$$

where: m - is mass flux source

ϕ_p - is the value of the dependent variable at point near the boundary

$C\phi$, $V\phi$ - two coefficients specified by the user. The source term for

mass flux is simply

$$S_m = C_m (V_m - P_p) \quad (7.4)$$

where: P_p -is the pressure near the boundary and C_m , V_m are two coefficients.

The values of C_ϕ and V_ϕ for the dependent variables in SATELLITE are: At the

Inlet:

$$C_m = 2 \frac{\gamma}{\gamma-1} \frac{1}{Wi} \quad (7.5)$$

$$V_m = P_o \rho_i / P_o \quad (7.6)$$

$$C_w = C_h = C_k = C_\epsilon = 0. \quad (7.7)$$

$$V_w = W_i \quad (7.8)$$

$$V_h = h_i \quad (7.9)$$

$$V_k = K_i \quad (7.10)$$

$$V_\epsilon = \epsilon_i \quad (7.11)$$

At the Outlet:

$$C_m = 1000 * W_i \cdot \rho_i / P_i \quad (7.12)$$

$$V_m = P_i \quad (7.13)$$

At the Wall (laminar)

$$C_w = \mu / (0.5 \Delta \mu_l) \quad (7.14)$$

$$V_w = 0 \quad (7.15)$$

$$C_h = \mu / Pr / (0.5 \cdot \Delta \mu_l) \quad (7.16)$$

$$V_h = C_p * T_w \quad (7.17)$$

At the Wall (turbulent)

$$C_w = C_h = C_k = C_\epsilon = WALL \quad (7.18)$$

$$V_w = V_k = V_\epsilon = 0 \quad (7.19)$$

$$V_h = C_p * T_w \quad (7.20)$$

8. Mesh Generation

In this work a two-dimensional mesh is being used with 18 x 29 cells in the y and z coordinate respectively. A Nonuniform grid has been used for both directions. Figures 3 and 4 shows the grid in the z direction. A finer grid is used in the blunt region, $IZ = (7 \div 17)$, and in the zone, where the inclined wall transitions to a straight wall, $IZ = (23 \div 26)$.

In the y coordinate, except in the boundary layer region, the grid is uniform. To obtain a finer grid resolution in the boundary layer for the laminar flow case the first five cells in the y direction from the wall obey the following proportionality relationship:

$$BYFRAC(IY) = \left(\frac{IY}{5}\right)^3 \left(\frac{\Delta_{max}}{10GH}\right) \quad (8.1)$$

Where $BYFRAC(IY)$ is the distance from the south side to the north side of the cell of particular interest, divided by total length of the domain, IY is the cell number, Δ_{max} is maximum allowable cell height, and GH is the half thickness of the TVC jet vane.

A fine grid resolution for the turbulent flow case is set up in the same way as laminar flow. The only difference comes from the selection of the first five cells in y direction. The following calculation shows the difference.

From the laminar solution and the given properties the following are known:

$$w = 885.2[m/s]$$

$$\mu_{lam} = 1 \cdot 10^{-5} [N.s/m]$$

$$Po = 5.5 \cdot 10^6 [Pa]$$

$$P_{static} = 1.048 \cdot 10^5 [Pa]$$

$$\gamma = 1.35$$

$$\rho = 1.835 \text{ [kg/m]}$$

Using the values above and the length of vane, which is 0.095m, A corresponding Reynolds number was calculated:

$$Re_z = \frac{\rho_{\infty} W_{\infty} Z}{\mu_{lam}} = \frac{(1.835 * 888.5 * 0.095)}{1 * 10^{-5}} = 1.54 * 10^6$$

Using a power law correlation for the boundary layer thickness:

$$\frac{\delta}{z} = 0.37 * Re_z^{-1/5} \quad (8.2)$$

From equation (8.2) the boundary layer thickness at the high end of the domain has been calculated as $\delta \approx 2 * 10^{-3} \text{ [m]}$

With Re based on W_{∞} the velocity parallel to the wall, $\frac{\Delta y}{2}$ the distance from the wall to the first grid node, ρ_{∞} the density, and μ_{lam} the laminar viscosity, Δy must satisfy the condition

$$Re_{\Delta} = \frac{\rho_{\infty} W_{\infty} \Delta y}{2 \mu_{lam}} > 132.25 \quad \text{or} \quad \Delta y > 6.48 * 10^{-6} \text{ [m]}$$

Therefore the interval of Δy is chosen such that

$$2 * 10^{-3} \text{ [m]} > \Delta y > 6.48 * 10^{-6} \text{ [m]}$$

In this effort using the relationship

$$BYFRAC(IY) = \left(\frac{IY}{5}\right)^2 \left(\frac{\Delta_{max}}{10GH}\right)$$

Δy has been calculated as $\Delta y = 8 * 10^{-5} \text{ [m]}$ which is in the required interval.

For both the laminar and turbulent cases, cells in the z direction were adjusted so that the points where possible physical phenomena such as shock waves and expansion fans are expected, very fine cells were used. In the other parts of the domain larger cells were used.

9. Heat Transfer Analysis

Skin friction and heat transfer quantities were calculated in both laminar and turbulent cases and they are shown in Figures (6 ÷ 11).

9.1 Laminar Calculation

In laminar flow fluxes can be derived directly from the gradients near the wall. The first cell is close "enough" to the wall and gradients of velocity and enthalpy do not change much in this region near the wall. The shear stress and heat flux in the laminar case will be:

$$\tau_w \approx \mu \frac{W_1}{\Delta Y_1/2} \quad (7.1)$$

$$q_w \approx \frac{\mu}{Pr} \frac{h_1 - h_w}{\Delta Y_1/2} \quad (7.2)$$

The skin friction coefficient and Stanton number will be:

$$C_f = \frac{2\tau_w}{\rho_\infty W_\infty^2} \quad (9.1)$$

$$S_t = q_w / [\rho_\infty u_\infty (h_r - h_w)] \quad (9.2)$$

where h_r is the recovery enthalpy

$$\frac{h_r}{h_o} = \frac{1 + \frac{r(\gamma-1)}{2} M_\infty^2}{1 + \frac{(\gamma-1)}{2} M_\infty^2} \quad (9.3)$$

r - is the recovery factor

$$r = \sqrt{Pr} \quad (\text{laminar flow}) \quad (9.4)$$

The coefficient of heat transfer in convection was calculated using

$$h_c = \rho_\infty U_\infty C_p S_t \quad (9.5)$$

9.2 Turbulent Calculations

In turbulent flow the gradients of velocity and enthalpy near the wall are very steep and change rapidly with distance from the wall.

Direct calculation of flux gradients is not accurate in this case. The log law approach is used to calculate skin friction. In the calculations using PHOENICS flow field, the following relation has been used.

$$C_f = \frac{2 \rho_w k_w}{w^2 \rho_\infty 3.33} \quad (9.6)$$

To obtain equation 9.6, the turbulent kinetic energy equation has been used as a starting point. [Ref. 5],

$$\begin{aligned} \rho \frac{Dk}{Dt} = \frac{\partial}{\partial y} \left(\frac{\mu_t}{\sigma_k} \frac{\partial k}{\partial y} \right) \\ + k \left[\frac{\mu_t}{k} \left(\frac{\partial u^2}{\partial y} \right) - C_D \frac{\rho^2 k}{\mu_t} \right] \end{aligned} \quad (9.7)$$

The source term of the turbulent kinetic energy equation should be zero near the wall which means

$$\frac{\mu_t}{k} \left(\frac{\partial u}{\partial y} \right)_w^2 - C_D \frac{\rho^2 k}{\mu_t} = 0 \quad (9.8)$$

therefore the shear stress on the wall can be defined as:

$$\tau_w = C_D^{1/2} \rho_w k_w \quad (9.9)$$

where k_w is the turbulent kinetic energy on the wall, ρ_w is the density on the wall and $C_D = 0.09$ [Ref. 5], substituting the values above into the Blasius skin friction relation the C_f equation becomes:

$$C_f = \frac{2 \tau_w}{\rho_\infty W_\infty^2} = \frac{\rho_w}{\rho_\infty} \frac{2}{W_\infty^2} \frac{k_w}{3.33} \quad (9.10)$$

The heat transfer quantities are evaluated from the Chilton-Colburn form of Reynolds analogy.

$$s_t = (C_f/2) * P_r^{-2/3} \quad (9.11)$$

$$q_w = s_t * \rho_\infty * U_\infty * (h_r - h_w) \quad (9.12)$$

where equation (9.3) is used to evaluate h_r with the recovery factor given as:

$$r = P_r^{1/3} \text{ (turbulent flow)} \quad (9.13)$$

The convective heat transfer coefficient is calculated by using equation (9.5)

10. Code and Computer

PHOENICS 81, Body Fitted Coordinate (BFC) version has been used in the computations (see Ref. 3). PHOENICS has been installed on NPS IBM 3033 MVS 1.3 computer. 400 sweeps per computer run provided a reasonable convergence in all runs except the turbulent blunt case continuity error of less than $4 \cdot 10^{-4}$ has been achieved in the three runs.

The continuity error is the total summation of the absolute mass imbalance in all cells divided by the inlet mass flux. CPU time consumption varies from case to case as follows:

Laminar Wedge	630	CPU Seconds
Turbulent Wedge	630	CPU Seconds
Laminar Blunt	630	CPU Seconds
Turbulent Blunt	1542	CPU Seconds for 1000 sweeps

11. Results and Discussion

The results of the calculations are available on appendix c. The tabular results include the values of pressure, velocities, enthalpy, temperature mach number, density, turbulent kinetic energy and rate of turbulent dissipation. The values are given in 18 x 29 cells points.

Skin friction and heat transfer results are shown in Figures (5-11). Laminar and turbulent skin friction and Stanton number in wedge flow show improvement compared to the results reported by Yukselen (Ref. 10). The lines are smoother and the oscillations at the end were eliminated. Basically the magnitudes are similar to those in Ref. 10.

Laminar blunt values are similar except near the beginning. The beginning, as expected in blunt zone, creates higher rates of heat transfer. Even though the blunt geometry used is a multi-wedge shape it should predict the correct values except for the stagnation point itself.

Turbulent blunt skin friction has different behavior. It has a very large value at the first point and then undershoots to values that are smaller than for wedge. It should also be kept in mind that the convergence of this case wasn't very successful.

12. Conclusions and Recommendations

1. PHOENICS was found to be a friendly code for simulating complicated mixed heat transfer fluid dynamics problems.

2. Derivation of heat transfer properties to a vane solid wall in laminar and turbulent flow has been installed in the code. It can be used for predictions of heat transfer rate in both cold and hot gas flow.

3. Two features have been added to the code in NPS: The restart option and the use of initial field, make it possible to simulate time dependent processes and solve the temperature variation in the vane itself.

LIST OF REFERENCES

1. Baldwin and McCormack, "Numerical Solution of the Interaction of a Strong Shock Wave with a Turbulent Boundary Layer," AIAA Paper 74-558, AIAA 7th Fluid and Plasma Dynamics Conference Palo Alto, Calif., June 17-19, 1974.
2. Shang, J. S., Hankey, W. L., and Law, C. H., "Numerical Simulation of Shock Wave-Turbulent Boundary Layer Interaction," AIAA Paper, 76-95, AIAA 14th Aerospace Sciences Meeting, Washington, D.C., January 1976.
3. Gunton, M. C., Rosten, H. L., and Spalding, D. B., Phoenix Instruction Manual, Spring 1983, CHAM Co, London.
4. White, Frank M., Viscous Fluid Flow, Mc Graw-Hill, Inc. 1974.
5. Launder, B., and Spalding, D. B., Lectures in Mathematical Models of Turbulence, Department of Mechanical Engineering Imperial College of Science and Technology, London, England, Academic Press, 1972.
6. Shapiro, Ascher H., The Dynamics and Thermodynamics of Compressible Fluid Flow, Volume, Department of Mechanical Engineering, Massachusetts Institute of Technology, New York, The Ronald Press Company.
7. Schlichting, Hermann, Engineering University of Braunschweig New York St. Louis, Boundary Layer Theory Mc Graw Hill Book Company, San Francisco Toronto, 1968.
8. Lin, C. C., Turbulent Flows and Heat Transfer, Princeton, New Jersey Princeton, New Jersey Princeton University Press, 1968.
9. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Mc Graw Hill, Inc., 1980.
10. Yukselen, A., Heat Transfer Modeling of Thrust Vector Control, Msc. Thesis, Naval Postgraduate School 1986.
11. Spalding D. B., "A General Computer Code for Two-Dimensional Elliptic Flows," Imperial College, London, 1977.

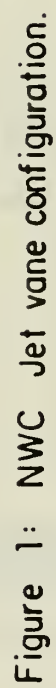


Figure 1: NWC Jet vane configuration.

All Dimensions in Millimeters

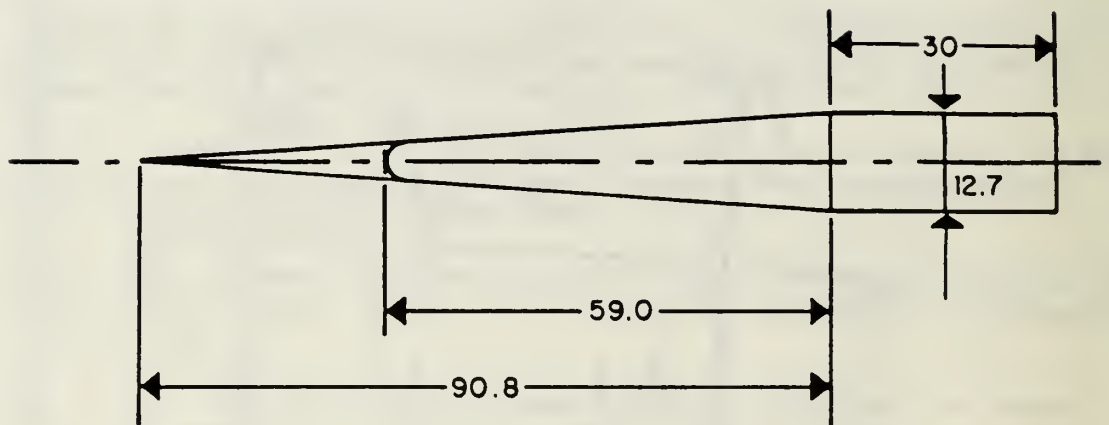


Figure 2: NWC Jet Vane Approximation

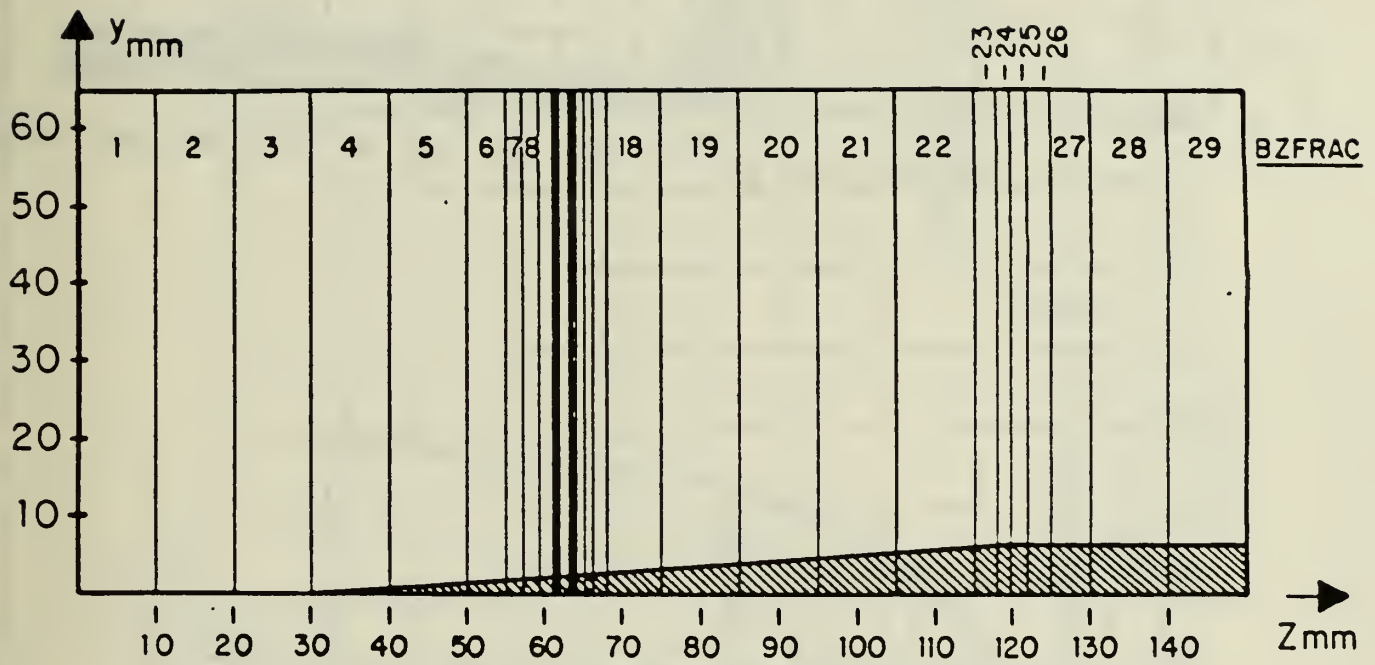


Figure 3: Wedge vane domain and grid

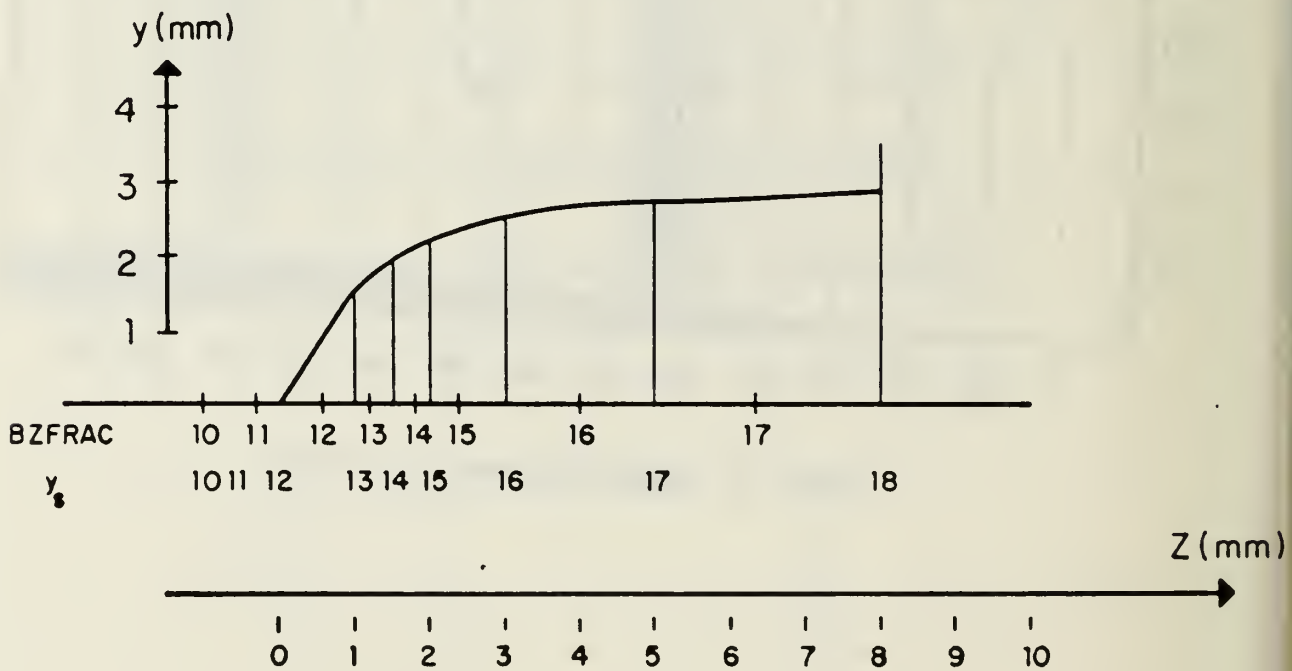
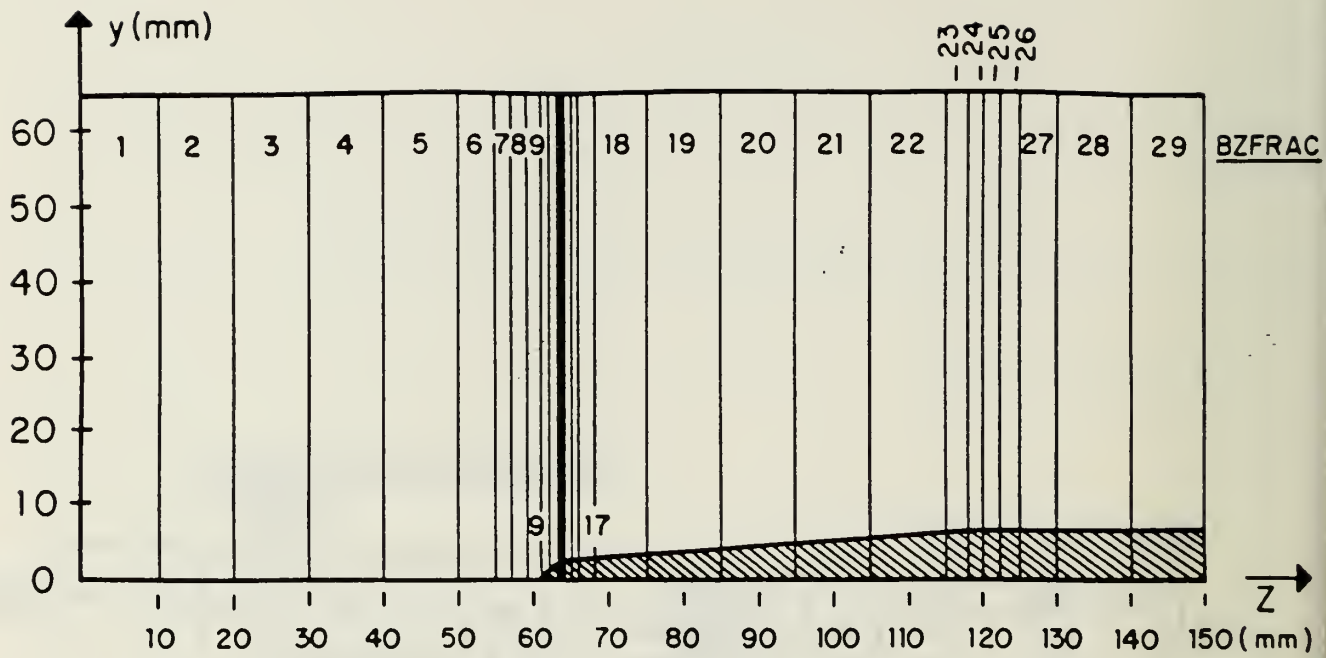


Figure 4: Blunt vane domain and grid.

RE NO.

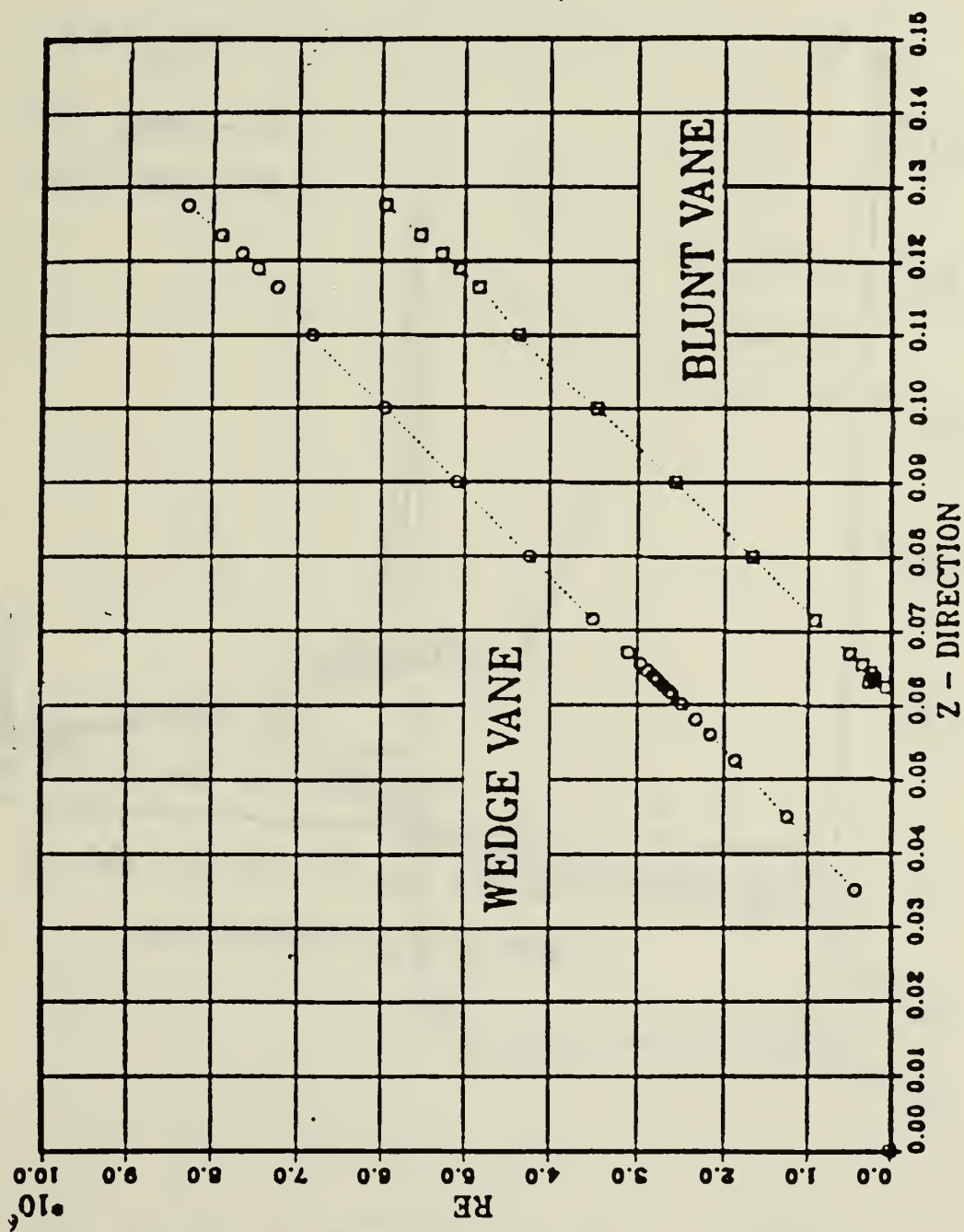


Figure 5: R_{ex} No. along the Vane

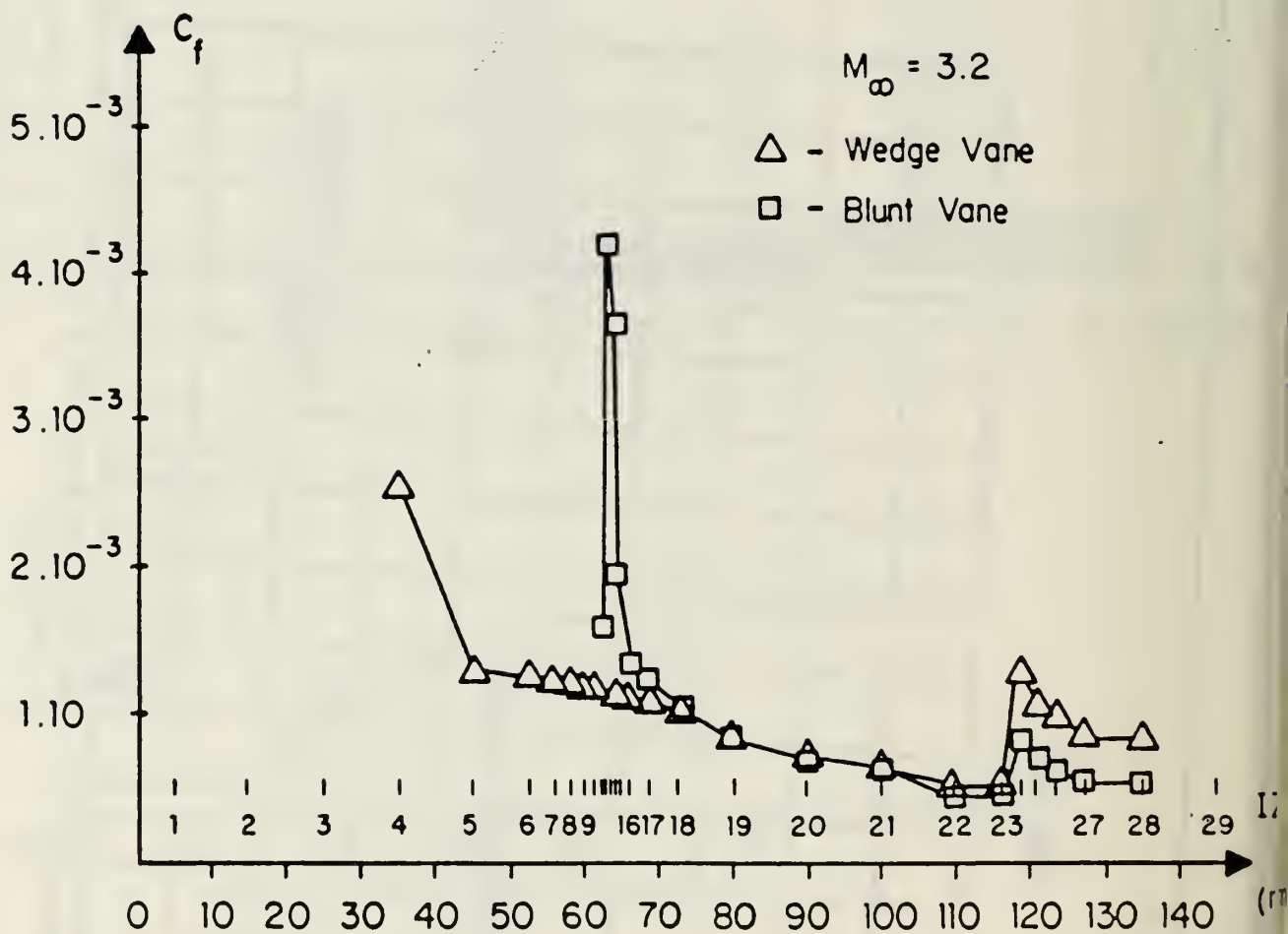


Figure 6: C_f in Laminar flow.

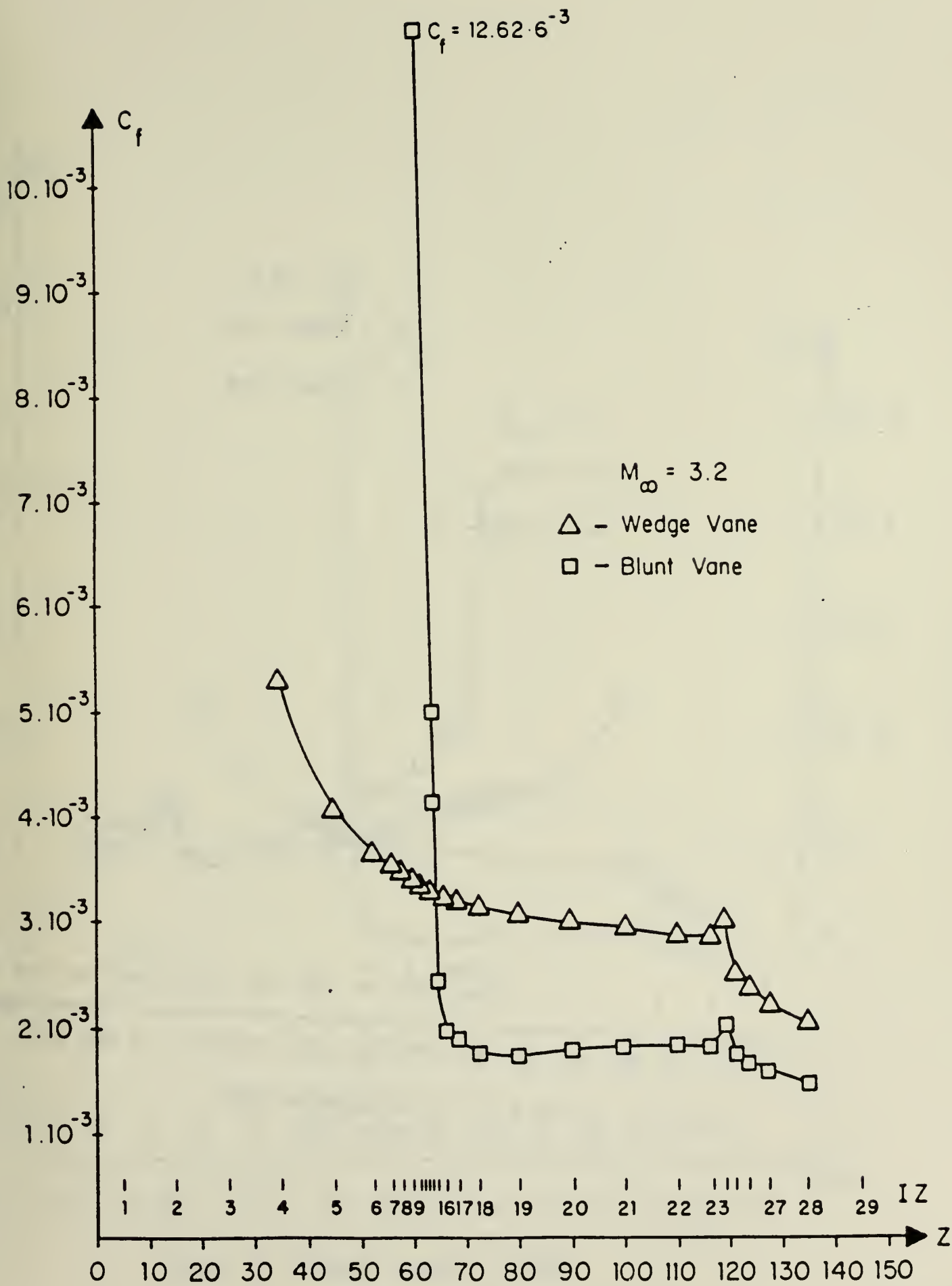


Figure 7: C_f in Turbulent flow

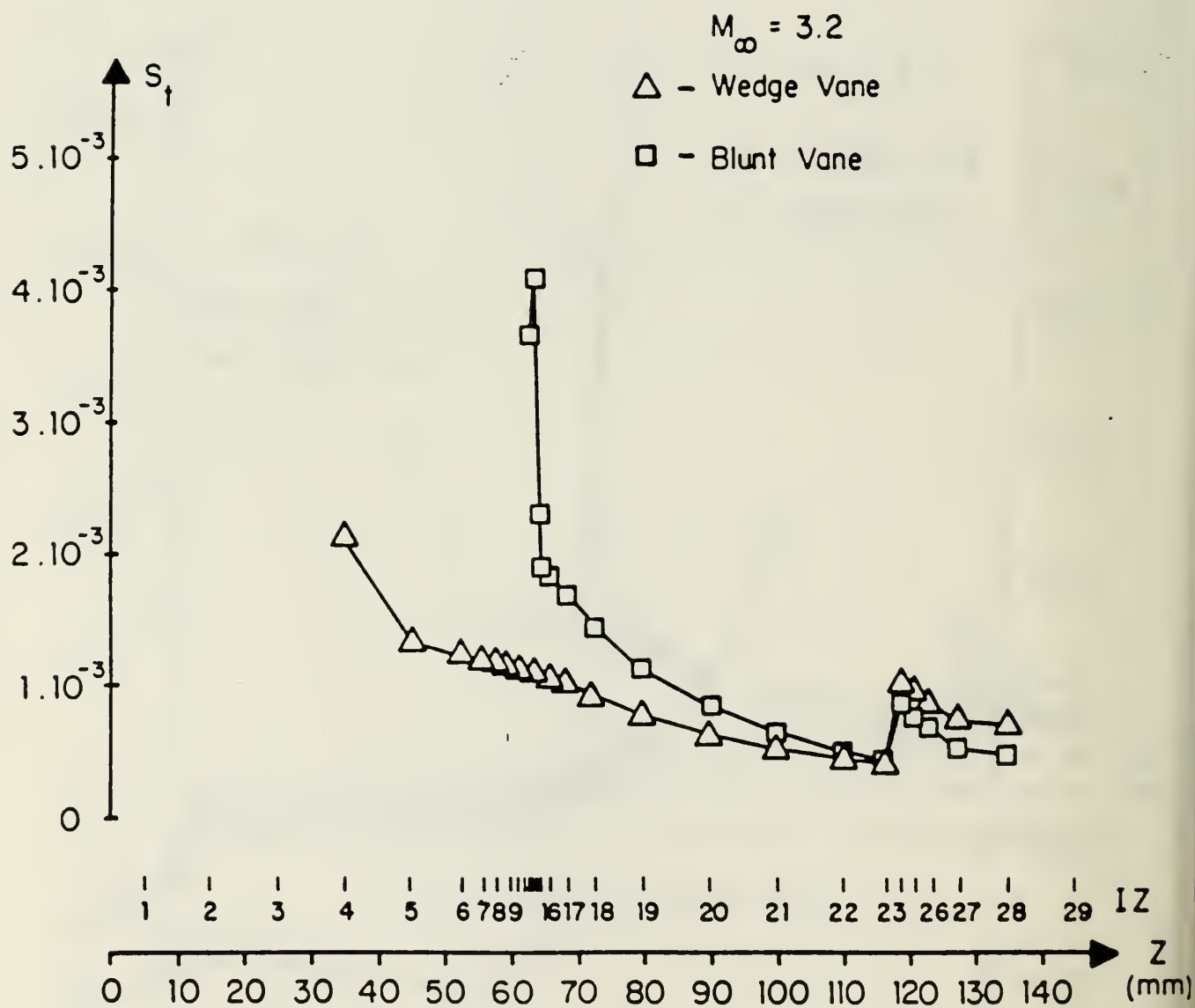


Figure 8: S_f in Laminar flow

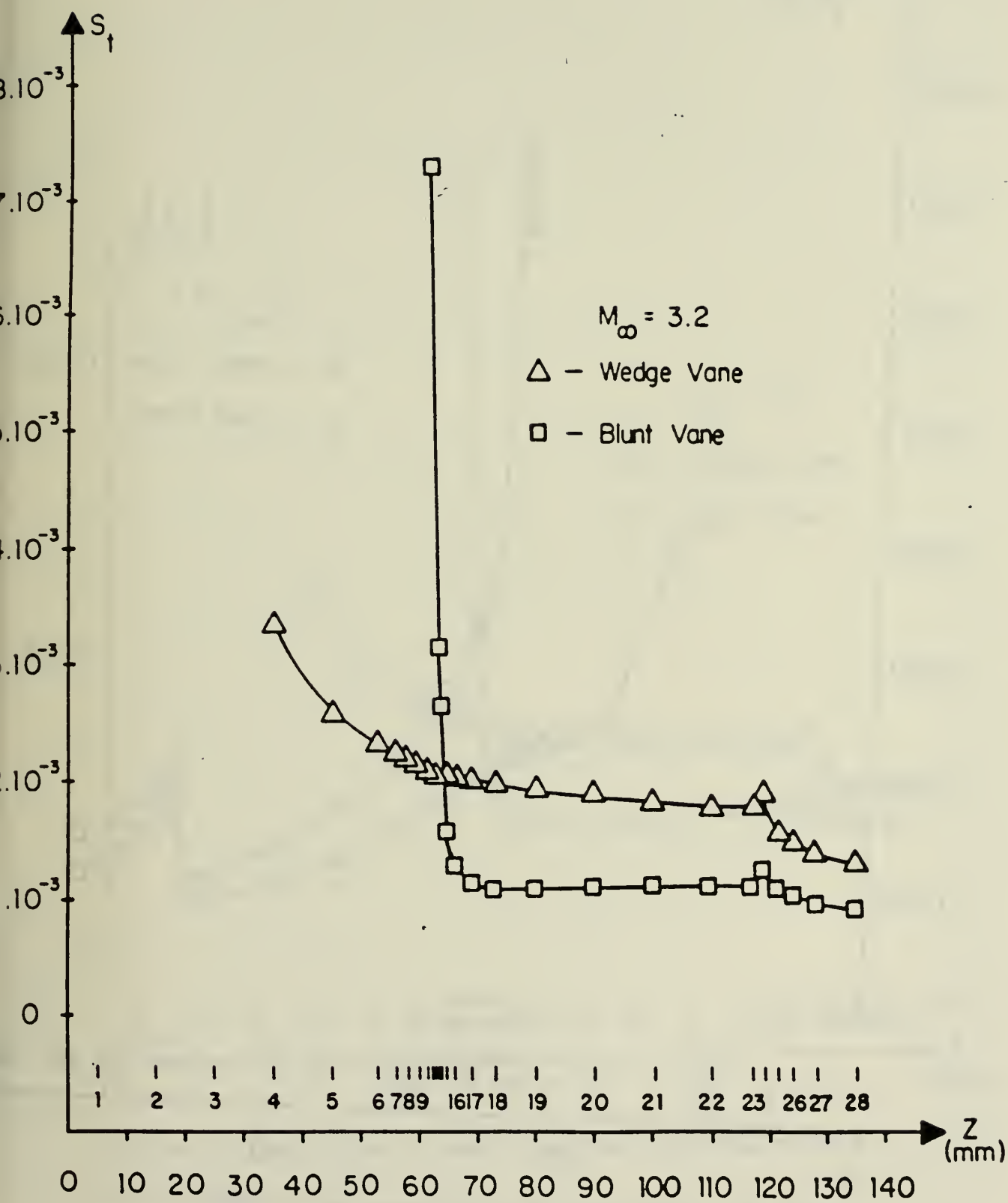


Figure 9: Turbulent stanton number.

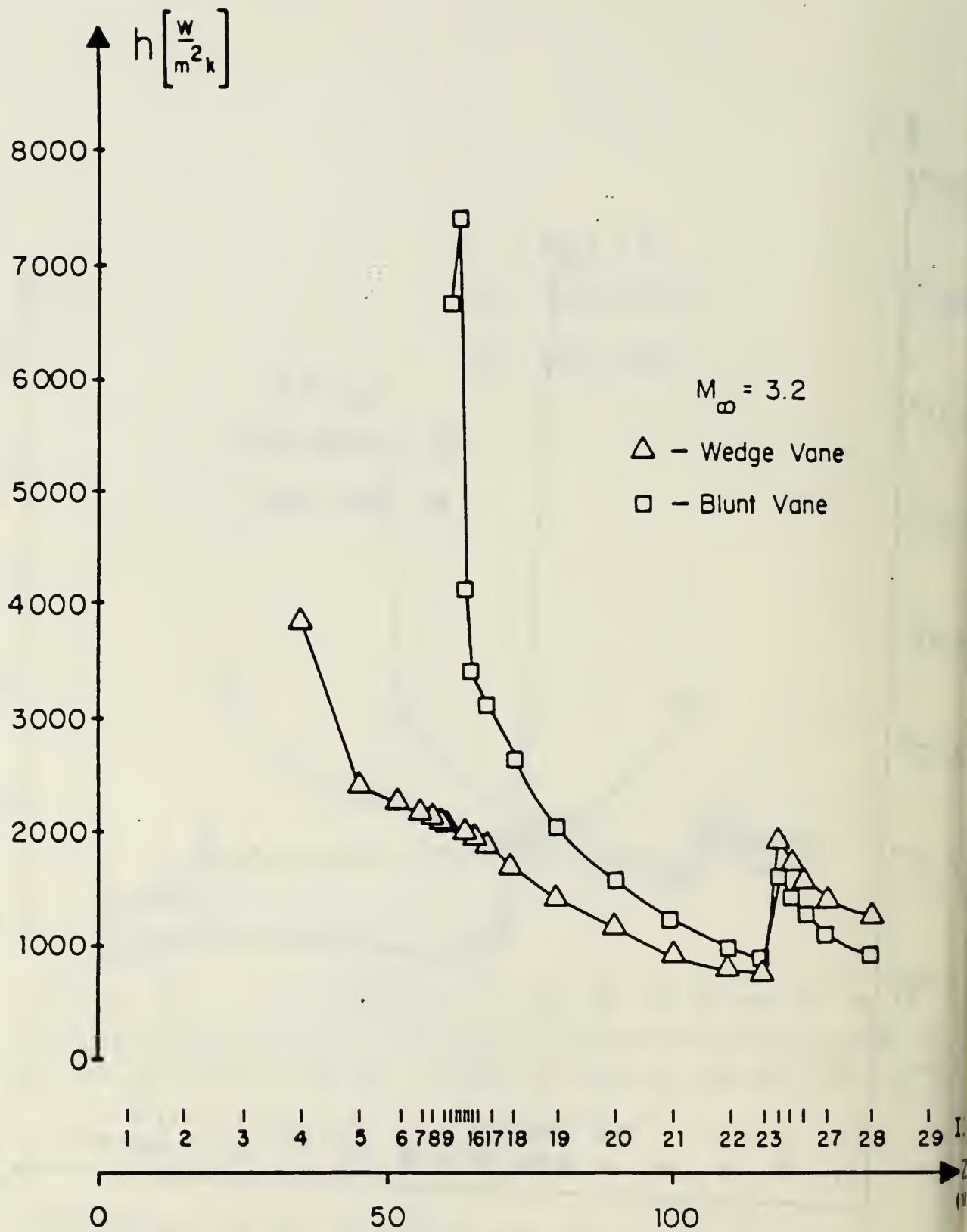


Figure 10: Coefficient of heat convection in laminar flow.

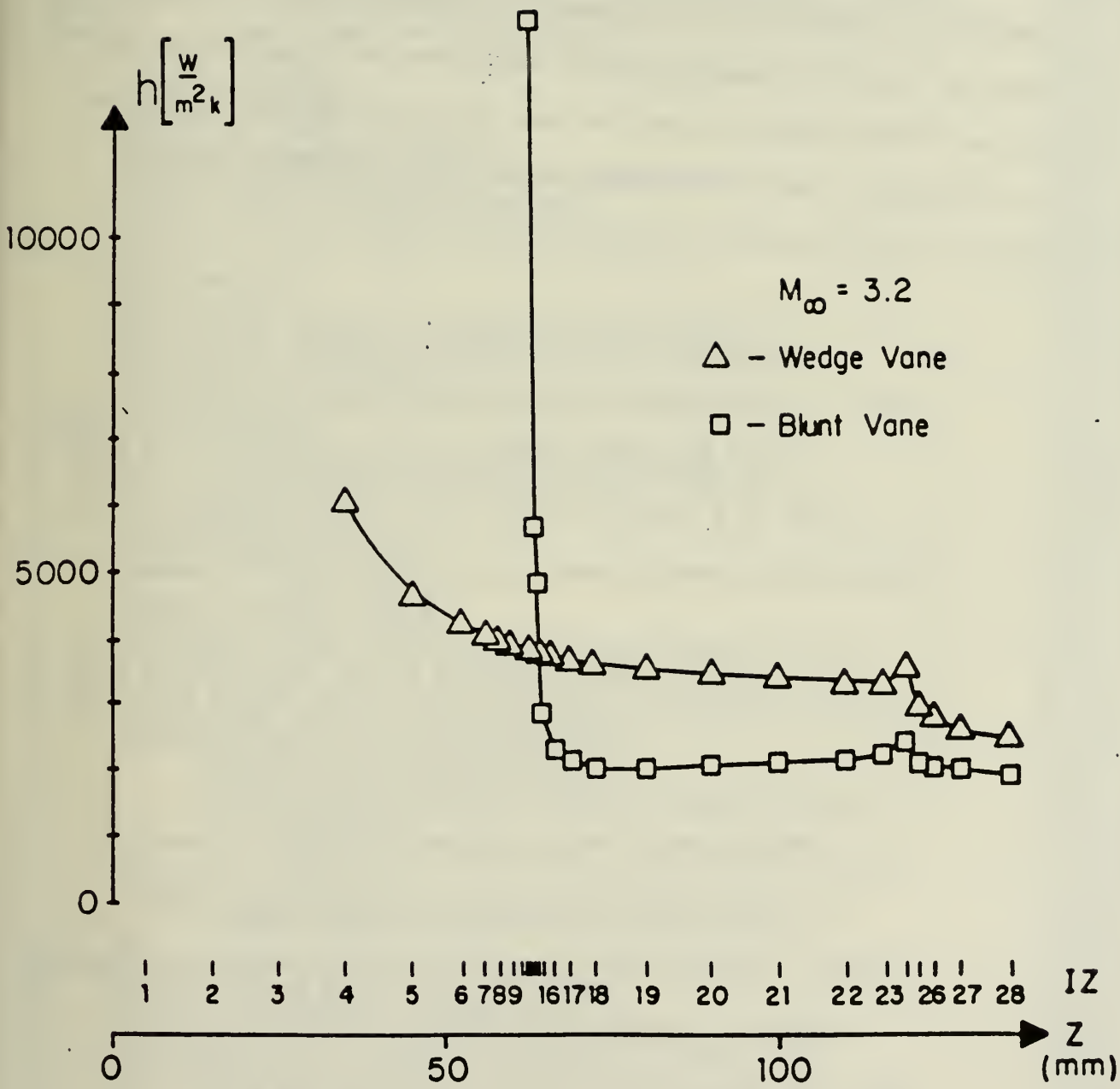


Figure 11: Coefficient of heat convection in turbulent flow.

Appendix A

Satellite Listing

Two subroutines SATELLITE and GROUND had to be changed and improved. The full list is enclosed in Appendix A and B.

VAN4SAT and VANTSAT are the laminar and turbulent SATELLITE subroutines, the first has the blunt geometry and the second has the wedge (it can be changed easily from wedge to blunt and vice versa) VAN4GRD and VANTGRD are exactly the same. They are the GROUND subroutines, VANTGRD is given in Appendix B.

C	*\$DIRECTIVE**SATLIT	AMI LEITHNER		VAN00010
C	LAMINAR SOLUTION FOR NWC5	NY=18 NZ=29 YN=GTH		VAN00020
C	LECSAT CONVERTED TO DIAMSAT			VAN00030
C	*FILE NAME: MODBFCST.FTN			VAN00040
C	*ABSTRACT: SATELLITE MODEL MAIN PROGRAM. THIS VERSION IS			VAN00050
C	FOR USE WITH THE BODY-FITTED COORDINATE SCHEME (SUMMER 1984			VAN00060
C	VERSION) PROVIDED AS AN ATTACHMENT TO SPRING 1983 PHOENICS.			VAN00070
C	*DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)			VAN00080
C	WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT			VAN00090
C	(SUMMER 1984).			VAN00100
C	*AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY			VAN00110
C	SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.			VAN00120
C	XXX STANDARD SECTION 1 STARTS:			VAN00130
C	-----			VAN00140
C	CHAPTER 1 COMMON BLOCKS AND USER'S DATA.			VAN00150
C	-----			VAN00160
	INCLUDE (CMNGUS)			VAN00170
	INCLUDE (CMNGRF)			VAN00180
	INCLUDE (GUSSEQ)			VAN00190
	COMMON/CPI/IPWRIT,IDUM(243)			VAN00200
	DIMENSION GDTAPE(3),DFAULT(4)			VAN00210
	DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)			VAN00220
	LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST			VAN00230
	INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE			VAN00240
	INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,			VAN00250
	&C3,C4			VAN00260
	REAL NORTH,LOW			VAN00270
	LOGICAL BFC			VAN00280
	EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)			VAN00290
	EQUIVALENCE (ARRAY3(1),SPARE1(1)),(M1,R1),(M2,R2)			VAN00300
	EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,LOGIC(88))			VAN00310
	EQUIVALENCE (LOGIC(20),BFC)			VAN00320
C	XXX STANDARD SECTION 1 ENDS.			VAN00330
C	*\$DIRECTIVE**CMNBFI\$\$			VAN00340
C	THIS FILE CONTAINS SATELLITE COMMON BLOCKS FOR BFC'S			VAN00350
C	F1 MUST BE DIMENSIONED TO GREATER THAN OR EQUAL TO			VAN00360
C	(NX+NY+17*NZ+24*NX*NY+6*(NX+1)*(NY+1)+6*ND). THE VALUE			VAN00370
C	OF THE DIMENSION MUST BE SET AS NBFC IN GROUP 6 OF SATLIT.			VAN00380
	COMMON/F0B/F1(5000)			VAN00390
	COMMON/CIB/ND/CIC/KOORD			VAN00400
	COMMON/CID/KDBGG,KDBGMF,KDBGCD,KDBIND,KDBMFX,KDBCDD,KDBPCS,			VAN00410
	& KDBGUV,KDBGPV			VAN00420
	COMMON/CIE/KDBGGS,KDBINS			VAN00430
	COMMON/CIF/IGEN/CIG/NCART			VAN00440
C	THE FOLLOWING ARRAYS MUST BE EXACTLY DIMENSIONED FOR NXPI,			VAN00450
C	NYP1 AND NZP1, BUT MAY BE OVER DIMENSIONED FOR ND.			VAN00460
C	THE BFRAC ARRAYS MUST BE DIMENSIONED TO ALLOW FOR SETTINGS			VAN00470
C	IN SATLIT, THEY MAY BE OVER DIMENSIONED.			VAN00480
	COMMON/CRA/XW(19,30,1)/CRB/XE(19,30,1)			VAN00490
	& /CRC/YS(2,30,1)/CRD/YN(2,30,1)			VAN00500
	& /CRE/ZL(2,19,1)/CRF/ZH(2,19,1)			VAN00510
	& /CRG/RCON/CRH/DARCY/CRI/BXFRAC(99)/CRJ/BYFRAC(99)			VAN00520
	& /CRK/BZFRAC(99)			VAN00530
	COMMON/CLA/STORSA(6),STORWD(6),STORP,STORPE,STORPN,			VAN00540
	& STORPH,STOR1,STOR2,STOR3,STOUNV,PRTBFC,STOCRN			VAN00550
	COMMON/CLC/BFPLLOT			VAN00560
	LOGICAL STORP,STORPE,STORPN,STORPH,STOR1,STOR2,STOR3,			VAN00570
	& STORSA,STORWD,STOUNV,PRTBFC,BFPLLOT,STOCRN			VAN00580
C	END			VAN00590
C	XXX USER SECTION 1 STARTS:			VAN00600
C	GRAFFIC ARRAYS DIMENSIONED AS NEEDED...			VAN00610
	COMMON/GRAF1/PHI1(1)/GRAF2/PHI2(1)			VAN00620
C	POROSITY & SPECIAL DATA ARRAYS DIMENSIONED AS NEEDED...			VAN00630
	DIMENSION PE(1,1,1),PN(1,1,1),PH(1,1,1),PC(1,1,1)			VAN00640
	DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)			VAN00650
C	USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.			VAN00660
	EQUIVALENCE(RAIR,RE(21)),(GAMA,RE(22)),(GSWP,RE(23))			VAN00670
	1,(GPR,RE(24)),(TW,RE(25)),(GEMUL,RE(26)),(JEMUL,INTGR(1))			VAN00680
C	USER PLACES HIS DATA STATEMENTS HERE.			VAN00690
	DATA NLSP,NISP,NRSP/1,1,1/			VAN00700
C	XXX USER SECTION 1 ENDS.			VAN00710
C	XXX STANDARD SECTION 2 STARTS:			VAN00720

```

C-----
CHAPTER 2  SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
C-----
C      PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C      STATEMENTS OF THIS CHAPTER.
C      DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
& 0.,1.,2.,3.,4.,5.,6.,7. /
C      DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
&C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
C      DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
C      DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
C      DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
C      DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
C      DATA NLDATA,NIDATA,NRDATA/309,194,421/
C      DATA NLCREG,NTCVRG/60,350/
C      DATA TITPP,TITC1,TITC2,TITC3/3HRHO,4HMACH,4HTEMP,4HCFST/
C      CALL TAPES(10,GDTAPE,3,1,4*NRDATA)
C-----
C      READ DEFAULT FILE IF BLOCKDATA ABSENT
      IF(INTGR1(29).NE.10) GO TO 2
      CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.      )
      GO TO 3
2 CALL DEFLT
CD 2 CALL TAPES(1,DFAULT,4,2,4*NRDATA)
CD CALL DATAIO(RD,1)
      CALL WRIT40(40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C)
3 CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED.      )
      LOGIC(89)=.TRUE.
C-----
CHAPTER 3  DEFINE DATA FOR NRUN RUNS.
C-----
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C--- GROUP 41MULTI-RUNS : RUN(1-30)<.T.,29*.F.>
C
      RUN(1)=.FALSE.
C NOTE: ALL RUNS ARE DEACTIVATED AT THIS POINT - USER SHOULD
C === SWITCH ON ONE ONLY OF RUNS 1-4 IN NEXT STATEMENT.
      RUN(1)=.TRUE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS:
      DO 10 IRUN=1,30
        IF(.NOT.RUN(IRUN)) GO TO 10
        NRUN=NRUN+1
        LSTRUN=IRUN
10 CONTINUE
      DO 999 IRUN=1,LSTRUN
        IF(.NOT.RUN(IRUN)) GO TO 999
        INTGR(11) = IRUN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS:
C--- ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C      TO 0.0, UNLESS OTHERWISE INDICATED.
C      E.G. BY VARIABLE<10>, OR <10.0> AS APPROPRIATE.
C      THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C      INDICATED, E.G. VARIABLE<.T.>, OR VARIABLE<.F.>.
C
C--- RUN1
C-----
C--- GROUP 1. FLOW TYPE :
C      PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
C-----
C--- GROUP 2. TRANSIENCE :
C      STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C      TLAST<1.E10>,TFRAC(1-30)<30*1.>
C      SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
C      CALL GRDPWR(0,NT,TLAST,POWER)
C-----
C--- GROUP 3. X-DIRECTION :
C      NX<1>,XULAST<1.0>,XFRAC(1-30)
C      SERVICE SUBROUTINE FOR POWER-LAW GRID:
C      CALL GRDPWR(1,NX,XULAST,POWER)
C-----

```

VAN00730
 VAN00740
 VAN00750
 VAN00760
 VAN00770
 VAN00780
 VAN00790
 VAN00800
 VAN00810
 VAN00820
 VAN00830
 VAN00840
 VAN00850
 VAN00860
 VAN00870
 VAN00880
 VAN00890
 VAN00900
 VAN00910
 VAN00920
 VAN00930
 VAN00940
 VAN00950
 VAN00960
 VAN00970
 VAN00980
 VAN00990
 VAN01000
 VAN01010
 VAN01020
 VAN01030
 VAN01040
 VAN01050
 VAN01060
 VAN01070
 VAN01080
 VAN01090
 VAN01100
 VAN01110
 VAN01120
 VAN01130
 VAN01140
 VAN01150
 VAN01160
 VAN01170
 VAN01180
 VAN01190
 VAN01200
 VAN01210
 VAN01220
 VAN01230
 VAN01240
 VAN01250
 VAN01260
 VAN01270
 VAN01280
 VAN01290
 VAN01300
 VAN01310
 VAN01320
 VAN01330
 VAN01340
 VAN01350
 VAN01360
 VAN01370
 VAN01380
 VAN01390
 VAN01400
 VAN01410
 VAN01420
 VAN01430
 VAN01440


```

C--- GROUP 4. Y-DIRECTION :
C NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWR(2,NY,YVLAST,POWER)
C NY=18
C-----
C--- GROUP 5. Z-DIRECTION :
C NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWR(3,NZ,ZWLAST,POWER)
C NZ=29
C-----
C--- GROUP 6. MOVING GRID OR DISTORTED (BODY-FITTED) GRID :
C --- MOVING GRID :
C MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2MIT
C-----
C --- BODY-FITTED GRID ---
C BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
C BXFRAC(1-NX)<1.0,NXM1*0.0>
C BYFRAC(1-NY)<1.0,NYM1*0.0>
C BZFRAC(1-NZ)<1.0,NZM1*0.0>
C SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1
C ONLY):
C CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
C XE(1-NYP1,1-NZP1,1-ND)<(NYP1*NZP1*ND)*1.0>,
C XW(1-NXP1,1-NZP1,1-ND),
C YN(1-NXP1,1-NZP1,1-ND)<(NXPI*NZP1*ND)*1.0>,
C YS(1-NXP1,1-NZP1,1-ND),
C ZH(1-NXP1,1-NYP1,1-ND)<(NXPI*NYP1*ND)*1.0>,
C ZL(1-NXP1,1-NYP1,1-ND),STORSA(1-6)<6*.F.>,STORWD(1-6)<6*.F.>,
C STORP<.F.>,STORPE<.F.>,STORPN<.F.>,STORPH<.F.>,STOUNV<.F.>,
C PRTBFC<.F.>,DARCY,BFPLT<.F.>
C CYCLIC BOUNDARY CONDITIONS ARE DEFAULTED INACTIVE ;
C TO ACTIVATE THEM AT SELECTED IZ SLABS USE SERVICE SUBROUTINE:
C CALL XCYIZ(IZ,.TRUE.)
C SERVICE SUBROUTINE TO DEACTIVATE CURVATURE TERMS IN U, V
C AND W EQUATIONS ASSOCIATED WITH CURVATURE OF IX, IY, IZ
C GRID LINES RESPECTIVELY:
C CALL UCURVE(IZ,.FALSE.)
C CALL VCURVE(IZ,.FALSE.)
C CALL WCURVE(IZ,.FALSE.)
C NCART<1>
C *WARNINGS|||||
C-----
C A) WHEN USING BFC'S STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
C AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
C COMPONENTS.
C B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C C) MOVING GRID,TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
C AVAILABLE WITH BFC OPTION.
C D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
C WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
C E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.
C *NOTES
C-----
C A) THE STANDARD VELOCITY-FIELD PRINTOUT FOR THE
C VELOCITY RESOLUTES IS ACTIVATED IN THE USUAL
C WAY. AN ADDITIONAL OPTION EXISTS FOR PRINTING THE
C CARTESIAN VELOCITY-COMPONENTS WHICH MAY BE
C ACTIVATED BY SETTING THE FOLLOWING LOGICALS:
C STOVAR(U2)=.T. FOR U-COMPONENT (CARTESIAN)
C STOVAR(V2)=.T. FOR V-COMPONENT (CARTESIAN)
C STOVAR(W2)=.T. FOR W-COMPONENT (CARTESIAN)
C SIMILARLY PRINTOUT OF NON-ORTHOGONAL VELOCITY
C COMPONENTS MAY BE ACTIVATED AS FOLLOWS:
C STOVAR(C4)=.T. FOR U-COMPONENT (NON-ORTHOG)
C STOVAR(H3)=.T. FOR V-COMPONENT (NON-ORTHOG)
C STOVAR(21)=.T. FOR W-COMPONENT (NON-ORTHOG)
C B) BFC (TO ACTIVATE THE BFC OPTION), IGEN (THE CODE FOR METHOD
C OF GRID SPECIFICATION), ND (NUMBER OF SUB-DOMAINS) AND
C NBFC (THE F1 ARRAY DIMENSION), MUST BE SET BEFORE
C "STANDARD BFC SECTION 2".
C=====

```

VAN01450
 VAN01460
 VAN01470
 VAN01480
 VAN01490
 VAN01500
 VAN01510
 VAN01520
 VAN01530
 VAN01540
 VAN01550
 VAN01560
 VAN01570
 VAN01580
 VAN01590
 VAN01600
 VAN01610
 VAN01620
 VAN01630
 VAN01640
 VAN01650
 VAN01660
 VAN01670
 VAN01680
 VAN01690
 VAN01700
 VAN01710
 VAN01720
 VAN01730
 VAN01740
 VAN01750
 VAN01760
 VAN01770
 VAN01780
 VAN01790
 VAN01800
 VAN01810
 VAN01820
 VAN01830
 VAN01840
 VAN01850
 VAN01860
 VAN01870
 VAN01880
 VAN01890
 VAN01900
 VAN01910
 VAN01920
 VAN01930
 VAN01940
 VAN01950
 VAN01960
 VAN01970
 VAN01980
 VAN01990
 VAN02000
 VAN02010
 VAN02020
 VAN02030
 VAN02040
 VAN02050
 VAN02060
 VAN02070
 VAN02080
 VAN02090
 VAN02100
 VAN02110
 VAN02120
 VAN02130
 VAN02140
 VAN02150
 VAN02160

```

C      ALL OTHER BFC DATA MUST BE SET AFTER "STANDARD BFC
C      SECTION 2.      =====
C      C) NXP1, NYP1, NZP1 STORE NX+1, NY+1, NZ+1; THESE ARE
C      AVAILABLE TO USER AFTER STANDARD BFC SECTION 2.
C      D) FOR IGEN=1 USE BXFRAC,BYFRAC & BZFRAC IN PLACE OF
C      XFRAC,YFRAC & ZFRAC.
C-----
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 STARTS:
C      DEFAULT SETTINGS:
C      NCART=10
C      BFC=.TRUE.
C      IGEN=1
C      ND=1
C      NBFC=5000
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 ENDS.:
C      *USER SETS BFC, IGEN, ND AND NBFC HERE:
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 STARTS:
C      CALL SB4I(NXP1,NX+1,NYP1,NY+1,NZP1,NZ+1,I,0)
C      IF(BFC) CALL BFCDFI(NBFC,XE,XW,YN,YS,ZH,ZL,ND,NXP1,NYP1,
C      & NZP1,NZ)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 ENDS.
C      *USER SETS ALL OTHER BFC VARIABLES HERE:
C      *USER NONIFORM GRID 1-8
C      GTH=65.E-3
C      GTL=150.E-3
C      GBETA=4.
C      GBETA=GBETA*3.1415927/180
C      GTAB=TAN(GBETA)
C      DELMAX=2.E-3
C      GNL=5.
C      GPWR=2.
C      DO 64 IY=1,5
C 64  BYFRAC(IY)=(FLOAT(IY)/GNL)**GPWR*DELMAX/GTH
C      BYFRAC(6)=BYFRAC(5)+3.E-3/GTH
C      DEL=(1.-BYFRAC(6))/(FLOAT(NY)-GNL-1)
C      DO 65 IY=7,NY
C 65  BYFRAC(IY)=BYFRAC(IY-1)+DEL
C-----ZZ-----
C      BZFRAC(1)=10.E-3
C      DO 66 IZ=2,5
C 66  BZFRAC(IZ)=10.E-3+BZFRAC(IZ-1)
C      BZFRAC(6)=BZFRAC(5)+5.E-3
C      DO 67 IZ=7,9
C 67  BZFRAC(IZ)=BZFRAC(IZ-1)+2.E-3
C      DO 68 IZ=10,10
C 68  BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
C      DO 77 IZ=11,14
C 77  BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
C      DO 78 IZ=15,15
C 78  BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
C      BZFRAC(16)=BZFRAC(15)+2.E-3
C      BZFRAC(17)=BZFRAC(16)+3.E-3
C      BZFRAC(18)=BZFRAC(17)+5.E-3
C      DO 69 IZ=19,22
C 69  BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
C      BZFRAC(23)=BZFRAC(22)+3.E-3
C      BZFRAC(24)=BZFRAC(23)+2.E-3
C      BZFRAC(25)=BZFRAC(24)+2.E-3
C      BZFRAC(26)=BZFRAC(25)+3.E-3
C      BZFRAC(27)=BZFRAC(26)+5.E-3
C      DO 71 IZ=28,NZ
C 71  BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
C      DO 72 IZ=1,NZ
C 72  BZFRAC(IZ)=BZFRAC(IZ)/GTL
C      CALL DOMAIN(1,1,NX,1,NY,1,NZ)
C      DO 61 IX=1,NXP1
C      DO 62 IY=1,NYP1
C      ZL(IX,IY,1)=0.0
C 62  ZH(IX,IY,1)=GTL
C      DO 63 IZ=1,NZP1
C      YN(IX,IZ,1)=GTH

```

VAN02170
 VAN02180
 VAN02190
 VAN02200
 VAN02210
 VAN02220
 VAN02230
 VAN02240
 VAN02250
 VAN02260
 VAN02270
 VAN02280
 VAN02290
 VAN02300
 VAN02310
 VAN02320
 VAN02330
 VAN02340
 VAN02350
 VAN02360
 VAN02370
 VAN02380
 VAN02390
 VAN02400
 VAN02410
 VAN02420
 VAN02430
 VAN02440
 VAN02450
 VAN02460
 VAN02470
 VAN02480
 VAN02490
 VAN02500
 VAN02510
 VAN02520
 VAN02530
 VAN02540
 VAN02550
 VAN02560
 VAN02570
 VAN02580
 VAN02590
 VAN02600
 VAN02610
 VAN02620
 VAN02630
 VAN02640
 VAN02650
 VAN02660
 VAN02670
 VAN02680
 VAN02690
 VAN02700
 VAN02710
 VAN02720
 VAN02730
 VAN02740
 VAN02750
 VAN02760
 VAN02770
 VAN02780
 VAN02790
 VAN02800
 VAN02810
 VAN02820
 VAN02830
 VAN02840
 VAN02850
 VAN02860
 VAN02870
 VAN02880

```

63  YS(IX,IZ,1)=0.0
C  YS(IX,13,1) SHOULD COME AFTER
    DO 662 IZ=16,25
CCC  DO 662 IZ=5,25
662  YS(IX,IZ,1)=(BZFRAC(IZ-1)-BZFRAC(3))*GTAB*GTL
    DO 663 IZ=13,15
        GZ12=(BZFRAC(IZ-1)-BZFRAC(11))*GTL
663  YS(IX,IZ,1)=SQRT(YS(IX,16,1)*GZ12*2.-GZ12**2)
    DO 664 IZ=26,NZ
664  YS(IX,IZ,1)=YS(IX,25,1)
61  CONTINUE
    STORSA(IFIX(LOW))=.TRUE.
    STORSA(IFIX(HIGH))=.TRUE.
    STORSA(IFIX(SOUTH))=.TRUE.
    STORWD(IFIX(SOUTH))=.TRUE.
    STORP=.TRUE.
    PRTBFC=.TRUE.
CDAR  DARCY=1.E10
C-----
C--- GROUP 7. BLOCKAGE: BLOCK<.F.>,IPLANE,IPWRIT
C  *SET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
C  CALL CONPOR(IR,TYPE,VALUE,IXF,IXL,IYF,IYL,IZF,IZL), WHERE:
C  IR=RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION; 'TYPE'= EAST,
C  WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE'=WANTED POROSITY
C  OVER REGION IXF,...IZL.
C  *DIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
C  PC(NX,NY,NZ) ABOVE.
C  *FOR FULLY-BLOCKED CELLS (IE. 'VALUE'= 0.0) USER NEED SET ONLY
C  THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
C  AUTOMATICALLY ZEROED.
C  *FOR SATELLITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
C  XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
C  *FOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
C  BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
C  IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
C  IN THIS CASE, THE USER M U S T SET A L L ELEMENTS OF
C  ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
C  CALL CR(PARRAY,VALUE,IXF,IXL,IYF,IYL,IZF,IZL,NX,NY,NZ)
C  ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
C  'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
C  *CONPOR M U S T N O T BE USED IN CONJUNCTION WITH EXPLICIT
C  SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).
C-----
C--- GROUP 8.DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
C  SOLVAR(1-25)<25*.F.>,STOVAR(1-25)<25*.F.>,CONC1(1-4)<4*.T.>
C  USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
C  P1,PP,U1,U2,V1,V2,W1,W2,M1,M2,RS,KE,EP,H1,H2,H3,C1,C2,C3,C4.
C  SOLVAR(P1)=.TRUE.
C  SOLVAR(PP)=.TRUE.
C  SOLVAR(V1)=.TRUE.
C  SOLVAR(W1)=.TRUE.
C  SOLVAR(H1)=.TRUE.
C  SOLVAR(KE)=.TRUE.
C  SOLVAR(EP)=.TRUE.
C  STOVAR(V2)=.TRUE.
C  STOVAR(W2)=.TRUE.
C  STOVAR(C1)=.TRUE.
C  STOVAR(C2)=.TRUE.
C  STOVAR(C3)=.TRUE.
C-----
C--- GROUP 9. VARIABLE LABELS :
C  TITLE(1-25)<2HP1,2HPP,2HU1,2HU2,2HV1,2HV2,2HW1,2HW2,2HR1,
C  2HR2,2HRS,2HKE,2HEP,2HH1,2HH2,2HH3,2HC1,2HC2,
C  2HC3,2HC4,2HRX,2HRY,2HRZ, 2*4H***>
C  TITLE(C1)=TITC1
C  TITLE(C2)=TITC2
C  TITLE(C3)=TITC3
C  TITLE(PP)=TITPP
C-----
C--- GROUP 10 PROPERTIES:
C  IRH01<1>,IRH02<1>,RH01<1.0>,RH02<1.0>,
C  ARH01<1.0>,BRH01<1.0>,CRH01<1.0>

```

VAN02890
 VAN02900
 VAN02910
 VAN02920
 VAN02930
 VAN02940
 VAN02950
 VAN02960
 VAN02970
 VAN02980
 VAN02990
 VAN03000
 VAN03010
 VAN03020
 VAN03030
 VAN03040
 VAN03050
 VAN03060
 VAN03070
 VAN03080
 VAN03090
 VAN03100
 VAN03110
 VAN03120
 VAN03130
 VAN03140
 VAN03150
 VAN03160
 VAN03170
 VAN03180
 VAN03190
 VAN03200
 VAN03210
 VAN03220
 VAN03230
 VAN03240
 VAN03250
 VAN03260
 VAN03270
 VAN03280
 VAN03290
 VAN03300
 VAN03310
 VAN03320
 VAN03330
 VAN03340
 VAN03350
 VAN03360
 VAN03370
 VAN03380
 VAN03390
 VAN03400
 VAN03410
 VAN03420
 VAN03430
 VAN03440
 VAN03450
 VAN03460
 VAN03470
 VAN03480
 VAN03490
 VAN03500
 VAN03510
 VAN03520
 VAN03530
 VAN03540
 VAN03550
 VAN03560
 VAN03570
 VAN03580
 VAN03590
 VAN03600


```

C      IEMU1<1>, EMU1<1.0>, EMULAM<1.E-10>      VAN03610
C      IHSAT, H1SAT, H2SAT, PSATEX<1.0>      VAN03620
C      SIGMA(1-25)<1.0, 2.0, 1., 1.E10, 1., 1.E10, 1., 1.E10,      VAN03630
C      4*1.0, 1.314, 1.0, 1.E10, 10*1.0>      VAN03640
C      IRH01=-1      VAN03650
C      PTOT=55.E5      VAN03660
C      TOT=555.55      VAN03670
C      RAIR=287.      VAN03680
C      GAMA=1.35      VAN03690
C      CP=RAIR/(1-1/GAMA)      VAN03700
C      TW=323.      VAN03710
C      HWALL=TW*CP      VAN03720
C      HTOT=CP*TOT      VAN03730
C      RHTOT=PTOT/TOT/RAIR      VAN03740
C      LOGIC(87)=.TRUE.      VAN03750
C      ARH01=RHTOT/PTOT**((1/GAMA)      VAN03760
C      BRH01=1./GAMA      VAN03770
C  TURBULENT OR LAMINAR      VAN03780
C      IEMU1=2      VAN03790
C      IEMU1=-1      VAN03800
C      JEMU1=IEMU1      VAN03810
C      EMU1=1.E-5      VAN03820
C      EMULAM=EMU1      VAN03830
C      GEMU1=EMU1      VAN03840
C      GPR=.7      VAN03850
C      SIGMA(24)=GPR      VAN03860
C      SIGMA(14)=.9      VAN03870
C-----      VAN03880
C---  GROUP 11 INTER-PHASE TRANSFER PROCESSES :      VAN03890
C      ICFIP, CFIPS, IMDOT, CMDOT, CA1I<1.E6>, CA2I<1.E6>      VAN03900
C-----      VAN03910
C---  GROUP 12 SPECIAL SOURCES :      VAN03920
C      ISPCSO(1-25), AGRAVX, AGRAVY, AGRAVZ, ABUOY, HREF      VAN03930
C-----      VAN03940
C---  GROUP 13 INITIAL FIELDS :      VAN03950
C      FIINIT(1-25)<25*1.E-10>      VAN03960
C  MACH NO. OF FREE STREAM      VAN03970
C      GMACH=3.2      VAN03980
C      A=1+(GAMA-1)/2*GMACH**2      VAN03990
C      TE=TOT/A      VAN04000
C      RHE=RHTOT/A**((1/(GAMA-1)))      VAN04010
C      PSTAT=PTOT/A**((GAMA/(GAMA-1)))      VAN04020
C      RH01=ARH01*PSTAT**BRH01      VAN04030
C      SONIC=SQRT(GAMA*RAIR*TE)      VAN04040
C      WIN=SONIC*GMACH      VAN04050
C      RKEIN=0.01*WIN**2      VAN04060
C      EPIN=0.16*RKEIN**1.5/GTH/2.      VAN04070
C      FIINIT(W1)=WIN      VAN04080
C      FIINIT(P1)=PSTAT      VAN04090
C      FIINIT(H1)=HTOT      VAN04100
C      FIINIT(KE)=RKEIN      VAN04110
C      FIINIT(EP)=EPIN      VAN04120
C-----      VAN04130
C---  GROUP 14 BOUNDARY/INTERNAL CONDITIONS :      VAN04140
C      ILOOP1, ILOOPN, XCYLE<.F.>, PBAR, REGION(1-10)<10*.T.>      VAN04150
C      *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE.. THE USER SHOULD      VAN04160
C      SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.      VAN04170
C      DO 14 I=1,10      VAN04180
C  14  REGION(I)=.FALSE.      VAN04190
C-----      VAN04200
C---  GROUP 15 TO 24; REGIONS 1 TO 10      VAN04210
C---  ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE      VAN04220
C      USER, PREFERABLY BY WAY OF:-      VAN04230
C      CALL PLACE(IREGN, TYPE, IXF, IXL, IYF, IYL, IZF, IZL) &      VAN04240
C      CALL COVAL(IREGN, VARBLE, COEFF, VALUE)      VAN04250
C      CALL PLACE(1, LOW, 1, NX, 1, NY, 1, 1)      VAN04260
C      CALL COVAL(1, M1, FIXFLU, WIN*RHE)      VAN04270
C  CDAR  CALL COVAL(1, M1, 1.E-20, 1.E+20*WIN*RHE)      VAN04280
C      GCM=2*GAMA/WIN/(GAMA-1)      VAN04290
C      GVM=PTOT*RHE/RHTOT      VAN04300
C      CALL COVAL(1, M1, GCM, GVM)      VAN04310
C      CALL COVAL(1, W1, ONLYMS, WIN)      VAN04320

```



```

      CALL COVAL(1,H1,ONLYMS,HTOT)
      CALL COVAL(1,KE,ONLYMS,RKEIN)
      CALL COVAL(1,EP,ONLYMS,EPIN)
      CALL PLACE(2,HIGH,1,NX,1,NY,NZ,NZ)
C     CALL COVAL(2,M1,FIXVAL,PSTAT*0.)
      CALL COVAL(2,M1,1000*WIN*RHE/PSTAT,PSTAT)
      CALL COVAL(2,H1,ONLYMS,HTOT)
C   WALL ALONG THE VANE IZ(11,NZ)
      GCM=EMUL/(.5*BYFRAC(1)*GTH)
      DY1=BYFRAC(1)*GTH
      GOEFF=EMUL/(0.5*DY1)
      GOEFH=EMUL/(0.5*DY1*SIGMA(24))
      CALL PLACE(3,SOUTH,1,NX,1,1,12,NZ)
C     CALL COVAL(3,W1,GOEFF,0.)
C     CALL COVAL(3,H1,GOEFH,HWALL)
      CALL COVAL(3,W1,WALL,0.)
      CALL COVAL(3,H1,WALL,HWALL)
      CALL COVAL(3,KE,WALL,0.)
      CALL COVAL(3,EP,WALL,0.)
C-----
C--- GROUP 25 GROUND STATION :
C   GROSTA<.F.>,NAMLIST<.F.>
C   *NAMLIST ACTIVATES NAMELIST IN GROUND.
C   GROSTA=.TRUE.
C-----
C--- GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS :
C   WHOLEP<.F.>,SUBPST<.F.>,DONACC<.F.>
C   WHOLEP=.TRUE.
C-----
C--- GROUP 27 SWEEP AND ITERATION NUMBERS :
C   FSWEPT<1>,LSWEPT<1>,LITHYD<1>,LITC<1>,LITKE<1>,LITH<1>,
C   LITER(1-25)<9*1,-1,15*1>
C   IVELF<1>,NVEL<1>,IVELL<10000>,
C   IKEF<1>,NKE<1>,IKEL<10000>,
C   IENTF<1>,NENT<1>,IENTL<10000>,
C   ICNCF<1>,NCNC<1>,ICNCL<10000>,
C   IRHO1F<1>,NRHO1<1>,IRHO1L<10000>,
C   IRHO2F<1>,NRHO2<1>,IRHO2L<10000>
C   LSWEPT=1201
C   GSWP=LSWEPT
C   FSWEPT=801
C   LITER(PP)=20
C   LITER(V1)=5
C   LITER(W1)=5
C   LITHYD=2
C-----
C--- GROUP 28 TERMINATION CRITERIA :
C   ENDIT(1-25)<9*1.E-10,0.5,15*1.E-10>
C   ENDIT(1)=1.E-5
C-----
C--- GROUP 29 RELAXATION :
C   RLXP<1>,>,RLXPXY<1>,>,RLXPZ<1>,>,RLXRHO<1>,>,RLXMDT<1>,>,
C   DTFALS(3-25)<23*1.E10>
C   DTFALS(W1)=1.E-5
C   DTFALS(V1)=1.E-5
C   DTFALS(KE)=1.E-5
C   DTFALS(EP)=1.E-6
C   RLXP=.3
C-----
C--- GROUP 30 LIMITS :
C   VELMAX<1.E10>,VELMIN<-1.E10>,RHOMAX<1.E10>,RHOMIN<1.E-10>,
C   TKEMAX<1.E10>,TKEMIN<1.E-10>,EMUMAX<1.E10>,EMUMIN<1.E-10>,
C   EPSMAX<1.E10>,EPSMIN<1.E-10>,AMDTMX<1.E10>,AMDTMN<-1.E10>
C   EPSMAX=1.E13
C-----
C--- GROUP 31 SLOWING DEVICES : SLORHO<1>,>,SLOEMU<1>,>
C   SLORHO=.2
C-----
C--- GROUP 32 PRINT-OUT OF VARIABLES :
C   PRINT(1-25)<.T.,.F.,23*.T.>,SUBWGR<.F.>
C   PRINT(C1)=.TRUE.
C   PRINT(C2)=.TRUE.

```

```

VAN04330
VAN04340
VAN04350
VAN04360
VAN04370
VAN04380
VAN04390
VAN04400
VAN04410
VAN04420
VAN04430
VAN04440
VAN04450
VAN04460
VAN04470
VAN04480
VAN04490
VAN04500
VAN04510
VAN04520
VAN04530
VAN04540
VAN04550
VAN04560
VAN04570
VAN04580
VAN04590
VAN04600
VAN04610
VAN04620
VAN04630
VAN04640
VAN04650
VAN04660
VAN04670
VAN04680
VAN04690
VAN04700
VAN04710
VAN04720
VAN04730
VAN04740
VAN04750
VAN04760
VAN04770
VAN04780
VAN04790
VAN04800
VAN04810
VAN04820
VAN04830
VAN04840
VAN04850
VAN04860
VAN04870
VAN04880
VAN04890
VAN04900
VAN04910
VAN04920
VAN04930
VAN04940
VAN04950
VAN04960
VAN04970
VAN04980
VAN04990
VAN05000
VAN05010
VAN05020
VAN05030
VAN05040

```

```

      PRINT(C3)=.TRUE.
      PRINT(PP)=.TRUE.
C-----
C--- GROUP 33 MONITOR PRINT-OUT :
C   IXMON<1>, IYMON<1>, IZMON<1>, NPRMON<1>, NPRMNT<1>
C   NPRMON=10
C   IYMON=2
C   IZMON=12
C-----
C--- GROUP 34 FIELD PRINT-OUT CONTROL :
C   NPRINT<100>, NTPRIN<100>, NXPRIN<1>, NYPRIN<1>, NZPRIN<1>,
C   IZPRF<1>, ISTPRF<1>, IZPRL<10000>, ISTPRL<10000>
C   NUMCLS<10>, KOUTPT
C   NPRINT=LSWEEP
C-----
C--- GROUP 35 TABLE CONTROL :
C   TABLES<.F.>, NTABLE, NTABVR, LINTAB, NPRTAB, NMON,
C   ITAB(1-8), MTABVR(1-8)
C-----
C   GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
C   MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
C--- GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP :
C   IZPR1<1>, IZPR2<1>, ISTPR1<1>, ISTPR2<1>
C-----
C--- GROUP 37 DEBUG SWEEP AND SUBROUTINES :
C   KEMU, KMAIN, KINDEX, KGEOM, KINPUT, KSODAT, KCOMPF, KSORCE,
C   KSOLV1, KSOLV2, KSOLV3, KCOMPV, KADJUST, KFLUX, KSHIFT, KDIF,
C   KCOMPV, KCOMPV, KCOMPW, KCOMPR, Kwall, KDBRHO<-1>, KDBEXP, KDBMDT
C   KDBGEN
C-----
C--- GROUP 38 MONITOR, TEST, AND FLAG :
C   MONITR<.F.>, FLAG<.F.>, TEST<.T.>, KFLAG<1>
C   END OF MAINTENANCE-ONLY SECTION
C-----
C--- GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
C   IERRP<1000>, RESREF(1,3-24)<25*1.>, RESMAP<.F.>,
C   RESID(1-25)<2*.F., 23*.T.>, KOUTPT
C   RESREF(1)=WIN*RHE
C   RESREF(7)=WIN*RESREF(1)
C   RESREF(5)=WIN*RESREF(1)*0.1
C   RESREF(H1)=HTOT*RESREF(1)
C   RESREF(KE)=RKEIN*RESREF(1)
C   RESREF(EP)=EPIN*RESREF(1)
C   IERRP=LSWEEP/20
C   KOUTPT=LSWEEP/20
C-----
C--- GROUP 40 SPECIAL DATA : LOGIC(1..10), INTGR(1..10), RE(21..30),
C   NLSP<1>, NISP<1>, NRSP<1>, SPDATA<.F.>, LSPDA(1), ISPDA(1), RSPDA(1)
C   USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C   TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C   SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
C   PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
C   ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C-----
C--- GROUP 42 RESTARTS AND DUMPS : SAVEM<.F.>, RESTRT<.F.>, KINPUT
C   SAVEM=.TRUE.
C   BFPLT=.TRUE.
C   RESTRT=.TRUE.
C-----
C--- GROUP 43 GRAFFIC :
C   GRAPHS<.F.>, ORTHOG<.T.>, ANTSYM, NPRT<1>, ITITL<5*4H***>
C--- FOR A GRAFFIC RUN, DIMENSION PHI1 & PHI2 AS FOLLOWS:
C   PHI1(NX*NY*NZ*NM)
C   PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)), WHERE
C   NM=NO. OF VARIABLES STORED + DENSITY(-IES)
C   IBLK=0 IF BLOCK=.FALSE., =4 IF A 3D RUN,
C   =3 IF A 2D.YZ RUN.
C-----
      IF(IRUN.EQ.1) GO TO 900
900 CONTINUE
C--- ALL RUNS

```

VAN05050
 VAN05060
 VAN05070
 VAN05080
 VAN05090
 VAN05100
 VAN05110
 VAN05120
 VAN05130
 VAN05140
 VAN05150
 VAN05160
 VAN05170
 VAN05180
 VAN05190
 VAN05200
 VAN05210
 VAN05220
 VAN05230
 VAN05240
 VAN05250
 VAN05260
 VAN05270
 VAN05280
 VAN05290
 VAN05300
 VAN05310
 VAN05320
 VAN05330
 VAN05340
 VAN05350
 VAN05360
 VAN05370
 VAN05380
 VAN05390
 VAN05400
 VAN05410
 VAN05420
 VAN05430
 VAN05440
 VAN05450
 VAN05460
 VAN05470
 VAN05480
 VAN05490
 VAN05500
 VAN05510
 VAN05520
 VAN05530
 VAN05540
 VAN05550
 VAN05560
 VAN05570
 VAN05580
 VAN05590
 VAN05600
 VAN05610
 VAN05620
 VAN05630
 VAN05640
 VAN05650
 VAN05660
 VAN05670
 VAN05680
 VAN05690
 VAN05700
 VAN05710
 VAN05720
 VAN05730
 VAN05740
 VAN05750
 VAN05760

CXXX USER SECTION 3 ENDS.	VAN05770
CXXX STANDARD SECTION 4 STARTS:	VAN05780
C-----	VAN05790
C WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...	VAN05800
IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)	VAN05810
IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)	VAN05820
IF(BFC) CALL WRTBFC(14,NBFC,XE,XW,YN,YS,ZH,ZL,	VAN05830
&ND,NX+1,NY+1,NZ+1,NZ,PRTBFC)	VAN05840
C OLD PRACTICES RETAINED FOR REFERENCE:	VAN05850
C IF(SPDATA) CALL SPCDAT(IRUN)	VAN05860
C IF(BLOCK) CALL PORDAT(IRUN)	VAN05870
IF(GRAPHS) CALL SORT(IRUN)	VAN05880
IF(RESTRT) GO TO 902	VAN05890
DO 901 INDVAR=1,25	VAN05900
IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901	VAN05910
CALL FLDDAT(IRUN)	VAN05920
GO TO 902	VAN05930
901 CONTINUE	VAN05940
902 CALL DATAIO(WRT,10)	VAN05950
IF(MONITR) CALL DATAIO(WRT,-6)	VAN05960
999 CONTINUE	VAN05970
STOP	VAN05980
END	VAN05990
C*** IGEN=1 SO BFCXYZ NOT REQUIRED.	VAN06000
C*** COMMENT OUT BOTH VERSIONS.	VAN06010
C-----	VAN06020
SUBROUTINE BFCXYZ (NXPI,NYPI,NZPI)	VAN06030
RETURN	VAN06040
END	VAN06050


```

C$DIRECTIVE**SATLIT  AMI LEITNER
C  LAMINAR SOLUTION FOR NWC5  NY=18 NZ=29 YN=GTH
C  LECSAT CONVERTED TO DIAMSAT
C  *FILE NAME: MODBFCST.FTN
C  *ABSTRACT: SATELLITE MODEL MAIN PROGRAM. THIS VERSION IS
C  FOR USE WITH THE BODY-FITTED COORDINATE SCHEME (SUMMER 1984
C  VERSION) PROVIDED AS AN ATTACHMENT TO SPRING 1983 PHOENICS.
C  *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C  WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C  (SUMMER 1984).
C  *AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY
C  SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C-----
CHAPTER 1  COMMON BLOCKS AND USER'S DATA.
C-----
      INCLUDE (CMNGUS)
      INCLUDE (CMNGRF)
      INCLUDE (GUSSEQ)
      COMMON/CPI/IPWRIT,IDUM(243)
      DIMENSION GDTAPE(3),DFAULT(4)
      DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)
      LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST
      INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE
      INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
&C3,C4
      REAL NORTH,LOW
      LOGICAL BFC
      EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)
      EQUIVALENCE (ARRAY3(1),SPARE1(1)),(M1,R1),(M2,R2)
      EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,LOGIC(88))
      EQUIVALENCE (LOGIC(20),BFC)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
C$DIRECTIVE**CMNBF1$$$
C  THIS FILE CONTAINS SATELLITE COMMON BLOCKS FOR BFC'S
C  F1 MUST BE DIMENSIONED TO GREATER THAN OR EQUAL TO
C  (NX+NY+17*NZ+24*NX*NY+6*(NX+1)*(NY+1)+6*ND). THE VALUE
C  OF THE DIMENSION MUST BE SET AS NBFC IN GROUP 6 OF SATLIT.
      COMMON/F0B/F1(5000)
      COMMON/CIB/ND/CIC/KOORD
      COMMON/CID/KDBGG,KDBGMF,KDBGCD,KDBIND,KDBMFX,KDBCDT,KDBPCS,
&      KDBGUV,KDBGPV
      COMMON/CIE/KDBGS,KDBINS
      COMMON/CIF/IGEN/CIG/NCART
C  THE FOLLOWING ARRAYS MUST BE EXACTLY DIMENSIONED FOR NXPI,
C  NYPI AND NZPI, BUT MAY BE OVER DIMENSIONED FOR ND.
C  THE BFRAC ARRAYS MUST BE DIMENSIONED TO ALLOW FOR SETTINGS
C  IN SATLIT, THEY MAY BE OVER DIMENSIONED.
      COMMON/CRA/XW(19,30,1)/CRB/XE(19,30,1)
&      /CRC/YS(2,30,1)/CRD/YN(2,30,1)
&      /CRE/ZL(2,19,1)/CRF/ZH(2,19,1)
&      /CRG/RCON/CRH/DARCY/CRI/BXFRAC(99)/CRJ/BYFRAC(99)
&      /CRK/BZFRAC(99)
      COMMON/CLA/STORSA(6),STORWD(6),STORP,STORPE,STORPN,
&      STORPH,STOR1,STOR2,STOR3,STOUNV,PRTBFC,STOGRN
      COMMON/CLC/BFPLLOT
      LOGICAL STORP,STORPE,STORPN,STORPH,STOR1,STOR2,STOR3,
&      STORSA,STORWD,STOUNV,PRTBFC,BFPLLOT,STOGRN
C  END
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C  GRAFFIC ARRAYS DIMENSIONED AS NEEDED...
      COMMON/GRAF1/PHI1(1)/GRAF2/PHI2(1)
C  POROSITY & SPECIAL DATA ARRAYS DIMENSIONED AS NEEDED...
      DIMENSION PE(1,1,1),PN(1,1,1),PH(1,1,1),PC(1,1,1)
      DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
C  USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      EQUIVALENCE(RAIR,RE(21)),(GAMA,RE(22)),(GSWP,RE(23))
      1,(GPR,RE(24)),(TW,RE(25)),(GEMU1,RE(26)),(JEMU1,INTGR(1))
C  USER PLACES HIS DATA STATEMENTS HERE.
      DATA NLSP,NISP,NRSP/1,1,1/
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:

```

```

VAN00010
VAN00020
VAN00030
VAN00040
VAN00050
VAN00060
VAN00070
VAN00080
VAN00090
VAN00100
VAN00110
VAN00120
VAN00130
VAN00140
VAN00150
VAN00160
VAN00170
VAN00180
VAN00190
VAN00200
VAN00210
VAN00220
VAN00230
VAN00240
VAN00250
VAN00260
VAN00270
VAN00280
VAN00290
VAN00300
VAN00310
VAN00320
VAN00330
VAN00340
VAN00350
VAN00360
VAN00370
VAN00380
VAN00390
VAN00400
VAN00410
VAN00420
VAN00430
VAN00440
VAN00450
VAN00460
VAN00470
VAN00480
VAN00490
VAN00500
VAN00510
VAN00520
VAN00530
VAN00540
VAN00550
VAN00560
VAN00570
VAN00580
VAN00590
VAN00600
VAN00610
VAN00620
VAN00630
VAN00640
VAN00650
VAN00660
VAN00670
VAN00680
VAN00690
VAN00700
VAN00710
VAN00720

```

```

C-----
CHAPTER 2  SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
C-----
C  PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C  STATEMENTS OF THIS CHAPTER.
C  DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
& 0.,1.,2.,3.,4.,5.,6.,7. /
C  DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
&C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
C  DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
C  DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
C  DATA WRT,RD,DFault/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
C  DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
C  DATA NLDATA,NIDATA,NRDATA/309,194,421/
C  DATA NLCREG,NTCVRG/60,350/
C  DATA TITPP,TITC1,TITC2,TITC3/3HRH0,4HMACH,4HTEMP,4HCFST/
C  CALL TAPES(10,GDTAPE,3,1,4*NRDATA)
C-----READ DEFAULT FILE IF BLOCKDATA ABSENT
IF(INTGR1(29).NE.10) GO TO 2
CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.      )
GO TO 3
2 CALL DEFLT
CD 2 CALL TAPES(1,DFault,4,2,4*NRDATA)
CD CALL DATAIO(RD,1)
CALL WRIT40(40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C)
3 CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED.      )
LOGIC(89)=.TRUE.
C-----
CHAPTER 3  DEFINE DATA FOR NRUN RUNS.
C-----
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C--- GROUP 41MULTI-RUNS : RUN(1-30)<.T.,29*.F.>
C
RUN(1)=.FALSE.
C NOTE: ALL RUNS ARE DEACTIVATED AT THIS POINT - USER SHOULD
C ==== SWITCH ON ONE ONLY OF RUNS 1-4 IN NEXT STATEMENT.
RUN(1)=.TRUE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS:
DO 10 IRUN=1,30
IF(.NOT.RUN(IRUN)) GO TO 10
NRUN=NRUN+1
LSTRUN=IRUN
10 CONTINUE
DO 999 IRUN=1,LSTRUN
IF(.NOT.RUN(IRUN)) GO TO 999
INTGR(11) = IRUN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS:
C--- ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C TO 0.0, UNLESS OTHERWISE INDICATED.
C E.G. BY VARIABLE<10>, OR <10.0> AS APPROPRIATE.
C THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C INDICATED, E.G. VARIABLE<.T.>, OR VARIABLE<.F.>.
C
C--- RUN1
C-----
C--- GROUP 1. FLOW TYPE :
C PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
C-----
C--- GROUP 2. TRANSIENCE :
C STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C TLAST<1.E10>,TFRAC(1-30)<30*1.>
C SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
C CALL GRDPWR(0,NT,TLAST,POWER)
C-----
C--- GROUP 3. X-DIRECTION :
C NX<1>,XULAST<1.0>,XFRAC(1-30)
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWR(1,NX,XULAST,POWER)
C-----

```

```

VAN00730
VAN00740
VAN00750
VAN00760
VAN00770
VAN00780
VAN00790
VAN00800
VAN00810
VAN00820
VAN00830
VAN00840
VAN00850
VAN00860
VAN00870
VAN00880
VAN00890
VAN00900
VAN00910
VAN00920
VAN00930
VAN00940
VAN00950
VAN00960
VAN00970
VAN00980
VAN00990
VAN01000
VAN01010
VAN01020
VAN01030
VAN01040
VAN01050
VAN01060
VAN01070
VAN01080
VAN01090
VAN01100
VAN01110
VAN01120
VAN01130
VAN01140
VAN01150
VAN01160
VAN01170
VAN01180
VAN01190
VAN01200
VAN01210
VAN01220
VAN01230
VAN01240
VAN01250
VAN01260
VAN01270
VAN01280
VAN01290
VAN01300
VAN01310
VAN01320
VAN01330
VAN01340
VAN01350
VAN01360
VAN01370
VAN01380
VAN01390
VAN01400
VAN01410
VAN01420
VAN01430
VAN01440

```



```

C--- GROUP 4. Y-DIRECTION :
C NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWR(2,NY,YVLAST,POWER)
C NY=18
C-----
C--- GROUP 5. Z-DIRECTION :
C NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
C SERVICE SUBROUTINE FOR POWER-LAW GRID:
C CALL GRDPWR(3,NZ,ZWLAST,POWER)
C NZ=29
C-----
C--- GROUP 6. MOVING GRID OR DISTORTED (BODY-FITTED) GRID :
C --- MOVING GRID :
C MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2M1T
C-----
C --- BODY-FITTED GRID ---
C BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KCOORD,RCON
C BXFRAC(1-NX)<1.0>,NXM1<0.0>
C BYFRAC(1-NY)<1.0>,NYM1<0.0>
C BZFRAC(1-NZ)<1.0>,NZM1<0.0>
C SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1
C ONLY):
C CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
C XE(1-NYP1,1-NZP1,1-ND)<(NYP1*NZP1*ND)*1.0>,
C XW(1-NYP1,1-NZP1,1-ND),
C YN(1-NXP1,1-NZP1,1-ND)<(NXP1*NZP1*ND)*1.0>,
C YS(1-NXP1,1-NZP1,1-ND),
C ZH(1-NXP1,1-NYP1,1-ND)<(NXP1*NYP1*ND)*1.0>,
C ZL(1-NXP1,1-NYP1,1-ND),STORSA(1-6)<6*.F.>,STORWD(1-6)<6*.F.>,
C STORP<.F.>,STORPE<.F.>,STORPN<.F.>,STORPH<.F.>,STOUNV<.F.>,
C PRTBFC<.F.>,DARCY,BFPL0T<.F.>
C CYCLIC BOUNDARY CONDITIONS ARE DEFAULTED INACTIVE ;
C TO ACTIVATE THEM AT SELECTED IZ SLABS USE SERVICE SUBROUTINE:
C CALL XCYIZ(IZ,.TRUE.)
C SERVICE SUBROUTINE TO DEACTIVATE CURVATURE TERMS IN U, V
C AND W EQUATIONS ASSOCIATED WITH CURVATURE OF IX, IY, IZ
C GRID LINES RESPECTIVELY:
C CALL UCURVE(IZ,.FALSE.)
C CALL VCURVE(IZ,.FALSE.)
C CALL WCURVE(IZ,.FALSE.)
C NCART<1>
C *WARNINGS|||||
C-----
C A) WHEN USING BFC'S STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
C AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
C COMPONENTS.
C B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C C) MOVING GRID,TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
C AVAILABLE WITH BFC OPTION.
C D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
C WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
C E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.
C *NOTES
C-----
C A) THE STANDARD VELOCITY-FIELD PRINTOUT FOR THE
C VELOCITY RESOLUTES IS ACTIVATED IN THE USUAL
C WAY. AN ADDITIONAL OPTION EXISTS FOR PRINTING THE
C CARTESIAN VELOCITY-COMPONENTS WHICH MAY BE
C ACTIVATED BY SETTING THE FOLLOWING LOGICALS:
C STOVAR(U2)=.T. FOR U-COMPONENT (CARTESIAN)
C STOVAR(V2)=.T. FOR V-COMPONENT (CARTESIAN)
C STOVAR(W2)=.T. FOR W-COMPONENT (CARTESIAN)
C SIMILARLY PRINTOUT OF NON-ORTHOGONAL VELOCITY
C COMPONENTS MAY BE ACTIVATED AS FOLLOWS:
C STOVAR(C4)=.T. FOR U-COMPONENT (NON-ORTHOG)
C STOVAR(H3)=.T. FOR V-COMPONENT (NON-ORTHOG)
C STOVAR(21)=.T. FOR W-COMPONENT (NON-ORTHOG)
C B) BFC (TO ACTIVATE THE BFC OPTION), IGEN (THE CODE FOR METHOD
C OF GRID SPECIFICATION), ND (NUMBER OF SUB-DOMAINS) AND
C NBFC (THE F1 ARRAY DIMENSION), MUST BE SET BEFORE
C "STANDARD BFC SECTION 2".
C=====

```

VAN01450
 VAN01460
 VAN01470
 VAN01480
 VAN01490
 VAN01500
 VAN01510
 VAN01520
 VAN01530
 VAN01540
 VAN01550
 VAN01560
 VAN01570
 VAN01580
 VAN01590
 VAN01600
 VAN01610
 VAN01620
 VAN01630
 VAN01640
 VAN01650
 VAN01660
 VAN01670
 VAN01680
 VAN01690
 VAN01700
 VAN01710
 VAN01720
 VAN01730
 VAN01740
 VAN01750
 VAN01760
 VAN01770
 VAN01780
 VAN01790
 VAN01800
 VAN01810
 VAN01820
 VAN01830
 VAN01840
 VAN01850
 VAN01860
 VAN01870
 VAN01880
 VAN01890
 VAN01900
 VAN01910
 VAN01920
 VAN01930
 VAN01940
 VAN01950
 VAN01960
 VAN01970
 VAN01980
 VAN01990
 VAN02000
 VAN02010
 VAN02020
 VAN02030
 VAN02040
 VAN02050
 VAN02060
 VAN02070
 VAN02080
 VAN02090
 VAN02100
 VAN02110
 VAN02120
 VAN02130
 VAN02140
 VAN02150
 VAN02160


```

C      ALL OTHER BFC DATA MUST BE SET AFTER "STANDARD BFC
C      SECTION 2.      =====
C      C) NXPI, NYPI, NZPI STORE NX+1, NY+1, NZ+1; THESE ARE
C      AVAILABLE TO USER AFTER STANDARD BFC SECTION 2.
C      D) FOR IGEN=1 USE BXFRAC, BYFRAC & BZFRAC IN PLACE OF
C      XFRAC, YFRAC & ZFRAC.
C-----
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 STARTS:
C  DEFAULT SETTINGS:
C  NCART=10
C  BFC=.TRUE.
C  IGEN=1
C  ND=1
C  NBFC=5000
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 ENDS.:
C  *USER SETS BFC, IGEN, ND AND NBFC HERE:
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 STARTS:
C  CALL SB4I(NXPI, NX+1, NYPI, NY+1, NZPI, NZ+1, I, 0)
C  IF(BFC) CALL BFCDFI(NBFC, XE, XW, YN, YS, ZH, ZL, ND, NXPI, NYPI,
C  & NZPI, NZ)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 ENDS.
C  *USER SETS ALL OTHER BFC VARIABLES HERE:
C  *USING NONIFORM GRID 1-8
C  GTH=65.E-3
C  GTL=150.E-3
C  GBETA=4.
C  GBETA=GBETA*3.1415927/180
C  GTAB=TAN(GBETA)
C  DELMAX=2.E-3
C  GNL=5.
C  GPWR=4.
C  DO 64 IY=1,5
64  BYFRAC(IY)=(FLOAT(IY)/GNL)*GPWR*DELMAX/GTH
C  BYFRAC(6)=BYFRAC(5)+3.E-3/GTH
C  DEL=(1.-BYFRAC(6))/(FLOAT(NY)-GNL-1)
C  DO 65 IY=7,NY
65  BYFRAC(IY)=BYFRAC(IY-1)+DEL
C-----ZZ-----
C  BZFRAC(1)=10.E-3
C  DO 66 IZ=2,5
66  BZFRAC(IZ)=10.E-3+BZFRAC(IZ-1)
C  BZFRAC(6)=BZFRAC(5)+5.E-3
C  DO 67 IZ=7,9
67  BZFRAC(IZ)=BZFRAC(IZ-1)+2.E-3
C  DO 68 IZ=10,11
68  BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
C  BZFRAC(12)=BZFRAC(11)+1.E-3
C  DO 77 IZ=13,14
77  BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
C  DO 78 IZ=15,15
78  BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
C  BZFRAC(16)=BZFRAC(15)+1.E-3
C  BZFRAC(17)=BZFRAC(16)+2.E-3
C  BZFRAC(18)=BZFRAC(17)+7.E-3
C  DO 69 IZ=19,22
69  BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
C  BZFRAC(23)=BZFRAC(22)+3.E-3
C  BZFRAC(24)=BZFRAC(23)+2.E-3
C  BZFRAC(25)=BZFRAC(24)+2.E-3
C  BZFRAC(26)=BZFRAC(25)+3.E-3
C  BZFRAC(27)=BZFRAC(26)+5.E-3
C  DO 71 IZ=28,NZ
71  BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
C  DO 72 IZ=1,NZ
72  BZFRAC(IZ)=BZFRAC(IZ)/GTL
C  CALL DOMAIN(1,1,NX,1,NY,1,NZ)
C  DO 61 IX=1,NXPI
C  DO 62 IY=1,NYPI
C  ZL(IX,IY,1)=0.0
62  ZH(IX,IY,1)=GTL
C  DO 63 IZ=1,NZPI

```

VAN02170
 VAN02180
 VAN02190
 VAN02200
 VAN02210
 VAN02220
 VAN02230
 VAN02240
 VAN02250
 VAN02260
 VAN02270
 VAN02280
 VAN02290
 VAN02300
 VAN02310
 VAN02320
 VAN02330
 VAN02340
 VAN02350
 VAN02360
 VAN02370
 VAN02380
 VAN02390
 VAN02400
 VAN02410
 VAN02420
 VAN02430
 VAN02440
 VAN02450
 VAN02460
 VAN02470
 VAN02480
 VAN02490
 VAN02500
 VAN02510
 VAN02520
 VAN02530
 VAN02540
 VAN02550
 VAN02560
 VAN02570
 VAN02580
 VAN02590
 VAN02600
 VAN02610
 VAN02620
 VAN02630
 VAN02640
 VAN02650
 VAN02660
 VAN02670
 VAN02680
 VAN02690
 VAN02700
 VAN02710
 VAN02720
 VAN02730
 VAN02740
 VAN02750
 VAN02760
 VAN02770
 VAN02780
 VAN02790
 VAN02800
 VAN02810
 VAN02820
 VAN02830
 VAN02840
 VAN02850
 VAN02860
 VAN02870
 VAN02880

```

      YN(IX,IZ,1)=GTH
      63  YS(IX,IZ,1)=0.0
C  YS(IX,13,1) SHOULD COME AFTER
      DO 662 IZ=5,25
CBL  DO 662 IZ=16,25
      662  YS(IX,IZ,1)=(BZFRAC(IZ-1)-BZFRAC(3))*GTAB*GTL
CBL  DO 663 IZ=13,15
CBL  GZ12=(BZFRAC(IZ-1)-BZFRAC(11))*GTL-.5E-3
CBL663 YS(IX,IZ,1)=SQRT(YS(IX,16,1)*GZ12*2.-GZ12**2)
      DO 664 IZ=26,NZ
      664  YS(IX,IZ,1)=YS(IX,25,1)
      61  CONTINUE
      STORSA(IFIX(LOW))=.TRUE.
      STORSA(IFIX(HIGH))=.TRUE.
      STORSA(IFIX(SOUTH))=.TRUE.
      STORWD(IFIX(SOUTH))=.TRUE.
      STORP=.TRUE.
      PRTBFC=.TRUE.
CDAR  DARCY=1.E10
C-----
C--- GROUP 7. BLOCKAGE: BLOCK<.F.>,IPLANE,IPWRIT
C  *SET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
C  CALL CONPOR(IR,TYPE,VALUE,IXF,IXL,IYF,IYL,IZF,IZL), WHERE:
C  IR=RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION; 'TYPE'= EAST,
C  WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE'=WANTED POROSITY
C  OVER REGION IXF,...IZL.
C  *DIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
C  PC(NX,NY,NZ) ABOVE.
C  *FOR FULLY-BLOCKED CELLS (IE. 'VALUE'= 0.0) USER NEED SET ONLY
C  THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
C  AUTOMATICALLY ZEROED.
C  *FOR SATELLITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
C  XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
C  *FOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
C  BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
C  IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
C  IN THIS CASE, THE USER M U S T SET A L L ELEMENTS OF
C  ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
C  CALL CR(PARRAY,VALUE,IXF,IXL,IYF,IYL,IZF,IZL,NX,NY,NZ)
C  ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
C  'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
C  *CONPOR M U S T N O T BE USED IN CONJUNCTION WITH EXPLICIT
C  SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).
C-----
C--- GROUP 8.DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
C  SOLVAR(1-25)<25*.F.>,STOVAR(1-25)<25*.F.>,CONC1(1-4)<4*.T.>
C  USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
C  P1,PP,U1,U2,V1,V2,W1,W2,M1,M2,RS,KE,EP,H1,H2,H3,C1,C2,C3,C4.
      SOLVAR(P1)=.TRUE.
      SOLVAR(PP)=.TRUE.
      SOLVAR(V1)=.TRUE.
      SOLVAR(W1)=.TRUE.
      SOLVAR(H1)=.TRUE.
CT  SOLVAR(KE)=.TRUE.
CT  SOLVAR(EP)=.TRUE.
      STOVAR(V2)=.TRUE.
      STOVAR(W2)=.TRUE.
      STOVAR(C1)=.TRUE.
      STOVAR(C2)=.TRUE.
      STOVAR(C3)=.TRUE.
C-----
C--- GROUP 9. VARIABLE LABELS :
C  TITLE(1-25)<2HP1,2HPP,2HU1,2HU2,2HV1,2HV2,2HW1,2HW2,2HR1,
C  2HR2,2HRS,2HKE,2HEP,2HH1,2HH2,2HH3,2HC1,2HC2,
C  2HC3,2HC4,2HRX,2HRY,2HRZ, 2*4H****>
      TITLE(C1)=TITC1
      TITLE(C2)=TITC2
      TITLE(C3)=TITC3
      TITLE(PP)=TITPP
C-----
C--- GROUP 10 PROPERTIES:
C  IRH01<1>,IRH02<1>,RH01<1.0>,RH02<1.0>,

```

```

VAN02890
VAN02900
VAN02910
VAN02920
VAN02930
VAN02940
VAN02950
VAN02960
VAN02970
VAN02980
VAN02990
VAN03000
VAN03010
VAN03020
VAN03030
VAN03040
VAN03050
VAN03060
VAN03070
VAN03080
VAN03090
VAN03100
VAN03110
VAN03120
VAN03130
VAN03140
VAN03150
VAN03160
VAN03170
VAN03180
VAN03190
VAN03200
VAN03210
VAN03220
VAN03230
VAN03240
VAN03250
VAN03260
VAN03270
VAN03280
VAN03290
VAN03300
VAN03310
VAN03320
VAN03330
VAN03340
VAN03350
VAN03360
VAN03370
VAN03380
VAN03390
VAN03400
VAN03410
VAN03420
VAN03430
VAN03440
VAN03450
VAN03460
VAN03470
VAN03480
VAN03490
VAN03500
VAN03510
VAN03520
VAN03530
VAN03540
VAN03550
VAN03560
VAN03570
VAN03580
VAN03590
VAN03600

```

```

C      ARH01<1.0>,BRH01<1.0>,CRH01<1.0>                                VAN03610
C      IEMU1<1>,EMU1<1.0>,EMULAM<1.E-10>                                VAN03620
C      IHSAT,H1SAT,H2SAT,PSATEX<1.0>                                    VAN03630
C      SIGMA(1-25)<1.0,2.0,1.,1.E10,1.,1.E10,1.,1.E10,          VAN03640
C      4*1.0,1.314,1.0,1.E10,10*1.0>                                VAN03650
C      IRH01=-1                                                        VAN03660
C      PTOT=55.E5                                                       VAN03670
C      TOT=555.55                                                       VAN03680
C      RAIR=287.                                                         VAN03690
C      GAMA=1.35                                                         VAN03700
C      CP=RAIR/(1-1/GAMA)                                              VAN03710
C      TW=323.                                                           VAN03720
C      HWALL=TW*CP                                                       VAN03730
C      HTOT=CP*TOT                                                       VAN03740
C      RHTOT=PTOT/TOT/RAIR                                              VAN03750
C      LOGIC(87)=.TRUE.                                                VAN03760
C      ARH01=RHTOT/PTOT**((1/GAMA).)                                  VAN03770
C      BRH01=1./GAMA                                                    VAN03780
C  TURBULENT OR LAMINAR                                              VAN03790
C      IEMU1=-1                                                         VAN03800
C      IEMU1=1                                                         VAN03810
C      JEMU1=IEMU1                                                      VAN03820
C      EMU1=1.E-5                                                       VAN03830
C      EMULAM=EMU1                                                      VAN03840
C      GEMU1=EMU1                                                       VAN03850
C      GPR=.7                                                            VAN03860
C      SIGMA(24)=GPR                                                    VAN03870
C      SIGMA(14)=GPR                                                    VAN03880
C-----
C--- GROUP 11 INTER-PHASE TRANSFER PROCESSES :                        VAN03890
C      ICFIP,CFIPS,IMDOT,CMDOT,CA1I<1.E6>,CA2I<1.E6>                VAN03900
C-----
C--- GROUP 12 SPECIAL SOURCES :                                       VAN03910
C      ISPCSO(1-25),AGRAVX,AGRAVY,AGRAVZ,ABUOY,HREF                 VAN03920
C-----
C--- GROUP 13 INITIAL FIELDS :                                       VAN03930
C      FIINIT(1-25)<25*1.E-10>                                       VAN03940
C      MACH NO. OF FREE STREAM                                       VAN03950
C      GMACH=3.2                                                        VAN03960
C      A=1+(GAMA-1)/2*GMACH**2                                         VAN03970
C      TE=TOT/A                                                         VAN03980
C      RHE=RHTOT/A**((1/(GAMA-1)))                                     VAN03990
C      PSTAT=PTOT/A**((GAMA/(GAMA-1)))                                VAN04000
C      RH01=ARH01*PSTAT**BRH01                                         VAN04010
C      SONIC=SQRT(GAMA*RAIR*TE)                                         VAN04020
C      WIN=SONIC*GMACH                                                  VAN04030
C      RKEIN=0.01*WIN**2                                               VAN04040
C      EPIN=0.16*RKEIN**1.5/GTH/2.                                     VAN04050
C      FIINIT(W1)=WIN                                                   VAN04060
C      FIINIT(P1)=PSTAT                                                VAN04070
C      FIINIT(H1)=HTOT                                                 VAN04080
C      FIINIT(KE)=RKEIN                                                VAN04090
C      FIINIT(EP)=EPIN                                                 VAN04100
C-----
C--- GROUP 14 BOUNDARY/INTERNAL CONDITIONS :                          VAN04110
C      ILOOP1,ILOOPN,XCYCLE<.F.>,PBAR,REGION(1-10)<10*.T.>          VAN04120
C      *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE.. THE USER SHOULD  VAN04130
C      SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.                VAN04140
C      DO 14 I=1,10                                                    VAN04150
C 14  REGION(I)=.FALSE.                                                VAN04160
C-----
C--- GROUP 15 TO 24; REGIONS 1 TO 10                                VAN04170
C--- ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE      VAN04180
C      USER, PREFERABLY BY WAY OF:-                                  VAN04190
C      CALL PLACE(IREGN,TYPE,IXF,IXL,IYF,IYL,IZF,IZL) &             VAN04200
C      CALL COVAL(IREGN,VARIABLE,COEFF,VALUE)                        VAN04210
C      CALL PLACE(1,LOW,1,NX,1,NY,1,1)                               VAN04220
C      CALL COVAL(1,M1,FIXFLU,WIN*RHE)                                VAN04230
C      CALL COVAL(1,M1,1.E-20,1.E+20*WIN*RHE)                       VAN04240
C      GCM=2*GAMA/WIN/(GAMA-1)                                         VAN04250
C      GVM=PTOT*RHE/RHTOT                                              VAN04260
C      CALL COVAL(1,M1,GCM,GVM)                                        VAN04270
C      CDAR                                                            VAN04280

```



```

      CALL COVAL(1,W1,ONLYMS,WIN)
      CALL COVAL(1,H1,ONLYMS,HTOT)
C     CALL COVAL(1,KE,ONLYMS,RKEIN)
C     CALL COVAL(1,EP,ONLYMS,EPIN)
      CALL PLACE(2,HIGH,1,NX,1,NY,NZ,NZ)
C     CALL COVAL(2,M1,FIXVAL,PSTAT*0.)
      CALL COVAL(2,M1,1000*WIN*RHE/PSTAT,PSTAT)
      CALL COVAL(2,H1,ONLYMS,HTOT)
C   WALL ALONG THE VANE IZ(11,NZ)
      GCM=EMU1/(.5*BYFRAC(1)*GTH)
      DY1=BYFRAC(1)*GTH
      GOEFF=EMU1/(0.5*DY1)
      GOEFH=EMU1/(0.5*DY1*SIGMA(24))
      CALL PLACE(3,SOUTH,1,NX,1,1,4,NZ)
C     CALL COVAL(3,W1,GOEFF,0.)
C     CALL COVAL(3,H1,GOEFH,HWALL)
      CALL COVAL(3,W1,WALL,0.)
      CALL COVAL(3,H1,WALL,HWALL)
CT    CALL COVAL(3,KE,WALL,0.)
CT    CALL COVAL(3,EP,WALL,0.)
C-----
C---  GROUP 25 GROUND STATION :
C     GROSTA<.F.>,NAMLIST<.F.>
C     *NAMLIST ACTIVATES NAMELIST IN GROUND.
C     GROSTA=.TRUE.
C-----
C---  GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS :
C     WHOLEP<.F.>,SUBPST<.F.>,DONACC<.F.>
C     WHOLEP=.TRUE.
C-----
C---  GROUP 27 SWEEP AND ITERATION NUMBERS :
C     FSWEPT<1>,LSWEPT<1>,LITHYD<1>,LITC<1>,LITKE<1>,LITH<1>,
C     LITER(1-25)<9*1,-1,15*1>
C     IVELF<1>,NVEL<1>,IVELL<10000>,
C     IKEF<1>,NKE<1>,IKEL<10000>,
C     IENTF<1>,NENT<1>,IENTL<10000>,
C     ICNCF<1>,NCNC<1>,ICNCL<10000>,
C     IRHO1F<1>,NRHO1<1>,IRHO1L<10000>,
C     IRHO2F<1>,NRHO2<1>,IRHO2L<10000>
      LSWEPT=400
      GSWP=LSWEPT
CR    FSWEPT=200
      LITER(PP)=20
      LITER(V1)=5
      LITER(W1)=5
      LITHYD=2
C-----
C---  GROUP 28 TERMINATION CRITERIA :
C     ENDIT(1-25)<9*1.E-10,0.5,15*1.E-10>
C     ENDIT(1)=1.E-5
C-----
C---  GROUP 29 RELAXATION :
C     RLXP<1>,RLXPXY<1>,RLXPZ<1>,RLXRHO<1>,RLXMDT<1>,
C     DTFALS(3-25)<23*1.E10>
      DTFALS(W1)=1.E-5
      DTFALS(V1)=1.E-5
      RLXP=.2
C-----
C---  GROUP 30 LIMITS :
C     VELMAX<1.E10>,VELMIN<-1.E10>,RHOMAX<1.E10>,RHOMIN<1.E-10>,
C     TKEMAX<1.E10>,TKEMIN<1.E-10>,EMUMAX<1.E10>,EMUMIN<1.E-10>,
C     EPSMAX<1.E10>,EPSMIN<1.E-10>,AMDTMX<1.E10>,AMDTMN<-1.E10>
C-----
C---  GROUP 31 SLOWING DEVICES : SLORHO<1>,SLOEMU<1>
      SLORHO=.2
C-----
C---  GROUP 32 PRINT-OUT OF VARIABLES :
C     PRINT(1-25)<.T.,.F.,23*.T.>,SUBWGR<.F.>
      PRINT(C1)=.TRUE.
      PRINT(C2)=.TRUE.
      PRINT(C3)=.TRUE.
      PRINT(PP)=.TRUE.

```

VAN04330
 VAN04340
 VAN04350
 VAN04360
 VAN04370
 VAN04380
 VAN04390
 VAN04400
 VAN04410
 VAN04420
 VAN04430
 VAN04440
 VAN04450
 VAN04460
 VAN04470
 VAN04480
 VAN04490
 VAN04500
 VAN04510
 VAN04520
 VAN04530
 VAN04540
 VAN04550
 VAN04560
 VAN04570
 VAN04580
 VAN04590
 VAN04600
 VAN04610
 VAN04620
 VAN04630
 VAN04640
 VAN04650
 VAN04660
 VAN04670
 VAN04680
 VAN04690
 VAN04700
 VAN04710
 VAN04720
 VAN04730
 VAN04740
 VAN04750
 VAN04760
 VAN04770
 VAN04780
 VAN04790
 VAN04800
 VAN04810
 VAN04820
 VAN04830
 VAN04840
 VAN04850
 VAN04860
 VAN04870
 VAN04880
 VAN04890
 VAN04900
 VAN04910
 VAN04920
 VAN04930
 VAN04940
 VAN04950
 VAN04960
 VAN04970
 VAN04980
 VAN04990
 VAN05000
 VAN05010
 VAN05020
 VAN05030
 VAN05040

```

C-----
C--- GROUP 33 MONITOR PRINT-OUT :
C    IXMON<1>, IYMON<1>, IZMON<1>, NPRMON<1>, NPRMNT<1>
C    NPRMON=5
C    IYMON=2
C    IZMON=12
C-----
C--- GROUP 34 FIELD PRINT-OUT CONTROL :
C    NPRINT<100>, NTPRIN<100>, NXPRIN<1>, NYPRIN<1>, NZPRIN<1>,
C    IZPRF<1>, ISTPRF<1>, IZPRL<10000>, ISTPRL<10000>
C    NUMCLS<10>, KOUTPT
C    NPRINT=LSWEEP
C-----
C--- GROUP 35 TABLE CONTROL :
C    TABLE<.F.>, NTABLE, NTABVR, LINTAB, NPRTAB, NMON,
C    ITAB(1-8), MTABVR(1-8)
C-----
C    GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
C    MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
C--- GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP :
C    IZPR1<1>, IZPR2<1>, ISTPR1<1>, ISTPR2<1>
C-----
C--- GROUP 37 DEBUG SWEEP AND SUBROUTINES :
C    KEMU, KMAIN, KINDEX, KGEOM, KINPUT, KSODAT, KCOMPF, KSORCE,
C    KSOLV1, KSOLV2, KSOLV3, KCOMPV, KADJST, KFLUX, KSHIFT, KDIF,
C    KCOMPU, KCOMPV, KCOMPW, KCOMPR, Kwall, KDBRHO<-1>, KDBEXP, KDBMDT
C    KDBGEN
C-----
C--- GROUP 38 MONITOR, TEST, AND FLAG :
C    MONITR<.F.>, FLAG<.F.>, TEST<.T.>, KFLAG<1>
C    END OF MAINTENANCE-ONLY SECTION
C-----
C--- GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
C    IERRP<1000>, RESREF(1,3-24)<25*1.>, RESMAP<.F.>,
C    RESID(1-25)<2*.F., 23*.T.>, KOUTPT
C    RESREF(1)=WIN*RHE
C    RESREF(7)=WIN*RESREF(1)
C    RESREF(5)=WIN*RESREF(1)*0.1
C    RESREF(H1)=HTOT*RESREF(1)
C    RESREF(KE)=RKEIN*RESREF(1)
C    RESREF(EP)=EPIN*RESREF(1)
C    IERRP=LSWEEP/10
C    KOUTPT=LSWEEP/10
C-----
C--- GROUP 40 SPECIAL DATA : LOGIC(1..10), INTGR(1..10), RE(21..30),
C    NLSP<1>, NISP<1>, NRSP<1>, SPDATA<.F.>, LSPDA(1), ISPDA(1), RSPDA(1)
C    USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C    TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C    SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
C    PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
C    ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C-----
C--- GROUP 42 RESTARTS AND DUMPS : SAVEM<.F.>, RESTRT<.F.>, KINPUT
C    SAVEM=.TRUE.
C    BFPLT=.TRUE.
C    RESTRT=.TRUE.
C-----
C--- GROUP 43 GRAFFIC :
C    GRAPHS<.F.>, ORTHOG<.T.>, ANTSYM, NPRT<1>, ITITL<5*4H****>
C--- FOR A GRAFFIC RUN, DIMENSION PHI1 & PHI2 AS FOLLOWS:
C    PHI1(NX*NY*NZ*NM)
C    PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)), WHERE
C    NM=NO. OF VARIABLES STORED + DENSITY(-IES)
C    IBLK=0 IF BLOCK=.FALSE., =4 IF A 3D RUN,
C    =3 IF A 2D.YZ RUN.
C-----
C    IF(IRUN.EQ.1) GO TO 900
C    900 CONTINUE
C--- ALL RUNS
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 4 STARTS:

```

VAN05050
 VAN05060
 VAN05070
 VAN05080
 VAN05090
 VAN05100
 VAN05110
 VAN05120
 VAN05130
 VAN05140
 VAN05150
 VAN05160
 VAN05170
 VAN05180
 VAN05190
 VAN05200
 VAN05210
 VAN05220
 VAN05230
 VAN05240
 VAN05250
 VAN05260
 VAN05270
 VAN05280
 VAN05290
 VAN05300
 VAN05310
 VAN05320
 VAN05330
 VAN05340
 VAN05350
 VAN05360
 VAN05370
 VAN05380
 VAN05390
 VAN05400
 VAN05410
 VAN05420
 VAN05430
 VAN05440
 VAN05450
 VAN05460
 VAN05470
 VAN05480
 VAN05490
 VAN05500
 VAN05510
 VAN05520
 VAN05530
 VAN05540
 VAN05550
 VAN05560
 VAN05570
 VAN05580
 VAN05590
 VAN05600
 VAN05610
 VAN05620
 VAN05630
 VAN05640
 VAN05650
 VAN05660
 VAN05670
 VAN05680
 VAN05690
 VAN05700
 VAN05710
 VAN05720
 VAN05730
 VAN05740
 VAN05750
 VAN05760

C-----		VAN05770
C WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...		VAN05780
IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)		VAN05790
IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)		VAN05800
IF(BFC) CALL WRTBFC(14,NBFC,XE,XW,YN,YS,ZH,ZL,		VAN05810
&ND,NX+1,NY+1,NZ+1,NZ,PRTBFC)		VAN05820
C OLD PRACTICES RETAINED FOR REFERENCE:		VAN05830
C IF(SPDATA) CALL SPCDAT(IRUN)		VAN05840
C IF(BLOCK) CALL PORDAT(IRUN)		VAN05850
IF(GRAPHS) CALL SORT(IRUN)		VAN05860
IF(RESTRT) GO TO 902		VAN05870
DO 901 INDVAR=1,25		VAN05880
IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901		VAN05890
CALL FLDDAT(IRUN)		VAN05900
GO TO 902		VAN05910
901 CONTINUE		VAN05920
902 CALL DATAIO(WRT,10)		VAN05930
IF(MONITR) CALL DATAIO(WRT,-6)		VAN05940
999 CONTINUE		VAN05950
STOP		VAN05960
END		VAN05970
C*** IGEN=1 SO BFCXYZ NOT REQUIRED.		VAN05980
C*** COMMENT OUT BOTH VERSIONS.		VAN05990
C-----		VAN06000
SUBROUTINE BFCXYZ (NXPI,NYPI,NZPI)		VAN06010
RETURN		VAN06020
END		VAN06030

Appendix B
Ground Listing

```

C$DIRECTIVE**MAIN      AMI LEITNER
C      LECGRD  LAST GEO. NZ=27 NY=18 LAMINAR FLOW
C      *FILE NAME: MODBFCGD.FTN
C      *INCLUDE DED SUBROUTINES: THE MODELS OF MAIN, GROUND & STRIDE.
C      *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C      WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C      (SUMMER 1984).
C      *SATELLITE FILE NAME: MODSTL.FTN
C      COMMON/ISHIFT/III(57),NFMAX
C SET F-ARRAY DIMENSION AS NEEDED, & SET NFMAX ACCORDINGLY.
C FOR BFC'S ALSO SET F1-ARRAY DIMENSION AS NEEDED ,AND SET
C NF1MAX ACCORDINGLY.
COMMON/F0B/F1(10000)
COMMON/NF0B/NF1MAX
COMMON F(25000)
NFMAX=25000
NF1MAX=10000
CALL MAIN1
STOP
END
C$DIRECTIVE**GROUND
SUBROUTINE GROUND(IRN,ICHAP,ISTP,ISWP,IZED,INDVAR)
INCLUDE (CMNGUS)
INCLUDE (GUSSEQ)
C      INCLUDE NMLIST
LOGICAL BFC
EQUIVALENCE (LOGIC(20),BFC)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C-----
C+++++MEANING OF SUBROUTINE ARGUMENTS:
C      IRN=RUN NUMBER; ICHAP=CHAPTER CALLED; ISTP=TIME STEP;
C      ISWP=SOLUTION SWEEP; IZED=Z-SLAB; INDVAR: SEE CHAPTERS BELOW.
C+++++USER-INTRODUCED VARIABLES & ARRAYS:
C      TO AVOID CONFLICT WITH VARIABLE NAMES USED IN COMMON, ALL
C      VARIABLES INTRODUCED BY THE USER SHOULD HAVE NAMES STARTING
C      WITH 'G' IF REAL, 'J' IF INTEGER, AND 'G' OR 'J' IF LOGICAL.
C      THUS GDZ(IZ) MIGHT BE A Z-INTERVAL ARRAY;
C      GW1(IY,IX) A 2-D ARRAY FOR AXIAL VELOCITY; ETC.
C      USER-GENERATED SUBROUTINES SHOULD BE NAMED CORRESPONDINGLY, EG
C      SUBROUTINE GVISC(GTEMP,GCNC,GVSC), FOR COMPUTING VISCOSITY
C      FROM CONCENTRATION & TEMPERATURE.
C+++++GROUND-TO-EARTH CONNECTING SUBROUTINES:
C      *USE GET(NAME,GARRAY,NY,NX) TO PUT VALUES OF VARIABLE NAMED
C      'NAME' INTO ARRAY 'GARRAY' DIMENSIONED GARRAY(NY,NX).
C      *USE SET(NAME,IXF,IXL,IYF,IYL,GARRAY,NY,NX) TO SET VARIABLE
C      'NAME' TO GARRAY(IY,IX) OVER THE REGION: IXF-IXL & IYF-IYL.
C      *USE PRNSLB(NAME) TO PRINT VARIABLE 'NAME' OVER X-Y PLANE.
C      *USE ADD(NAME,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C      TO ADD SOURCE TO VARIABLE NAMED 'NAME' (SEE CHAPTER 5).
C      *USE READIZ(IZED) IN CHAPTERS 1, 2, 8, & 9 TO ACCESS P1,...DM
C      & VOL,...AHDZ. (SEE FOOTNOTE TO LEGALITY TABLE)
C      *USE GET1D(NAME,GARRAY,NDIM) TO PUT VARIABLE NAMED 'NAME' IN
C      ONE-D ARRAY 'GARRAY' DIMENSIONED NDIM, THUS:
C      CALL GET1D(NAME,GNX,NX) FOR XG,...DXG & DIMENSION GNX(NX);
C      CALL GET1D(NAME,GNV,NY) FOR YG,...RV & DIMENSION GNV(NY);
C      CALL GET1D(NAME,GNZ,NZ) FOR ZG,...WGRID & DIMENSION GNZ(NZ).
C+++++LEGALITY TABLE FOR USE OF EARTH-CONNECTING SUBROUTINES:
C      ENTRIES IN TABLE GIVE CHAPTERS IN WHICH SUBROUTINES CAN BE
C      USED FOR VARIABLES IN LEFT-HAND COLUMN. (SUBROUTINE
C      STRIDE IS REGARDED AS BEING IN CHAPTER 3)
C-----
C      : VARIABLE: : GET & : SET : ADD : READIZ : GET1D :
C      :           : PRNSLB : : : : : :
C-----
C      :P1 - RZ : : ALL : 6 & 7 : 5 : 1,2,8,9 : NONE :
C      :P10 - RZH : : 3-7, 10-16 : 3 : NONE : NONE : NONE :
C      :VOL -AHDZ : : ALL : 3 : NONE : 1,2,8,9 : NONE :
C      :D1DP : : NONE : 10 : NONE : NONE : NONE :
C      :D2DP : : NONE : 11 : NONE : NONE : NONE :
C      :MU1,MU1H : : 5,13-16 : 12 : NONE : NONE : NONE :
C      :EXCO(L,H) : : NONE : 13 : NONE : NONE : NONE :
C      :CFP : : 5 : 14 : NONE : NONE : NONE :

```

```

C      :MDT      ::      5      :      15      :      NONE      :      NONE      :      NONE      :
C      :HST1,HST2::      5 & 15 :      16      :      NONE      :      NONE      :      NONE      :
C      :XG -WGRID::      NONE      :      NONE      :      NONE      :      NONE      :      ALL      :
C      -----
C      NOTES ON ABOVE TABLE:
C      *IN CHAPTERS 1, 2, 8, & 9 VARIABLES P1...DM & GEOMETRY
C      VOL...AHDZ CAN BE ACCESSED BUT ONLY IN CONJUNCTION WITH
C      USE OF READIZ, THUS:
C      DO 1 IZED=1,NZ
C      CALL READIZ(IZED)
C      1 CALL GET(... AS REQUIRED...)
C      *GEOMETRY ACCESSED BY READIZ IS THAT AT INITIAL TIME.
C      *D1DP & D2DP ONLY ACCESSIBLE IN UNSTEADY FLOWS.
C      +++++GROUND SERVICE SUBROUTINES:
C      *USE CONTUR(NAME,IPLANE,ILOC,NINT,I1,I2,J1,J2,GARRAY,NDIM) FOR
C      LINE-PRINTER PLOTS OF CONTOURS. 'NAME' = U1,...C4;
C      'IPLANE'= XPLANE, YPLANE, OR ZPLANE; ILOC SETS IX, IY, OR
C      IZ LOCATION OF IPLANE; I1, I2, J1, & J2 SET FIRST & LAST
C      CELLS IN HORIZ. & VERT. ON PLOT; GARRAY IS 1-D WORKING ARRAY
C      OF DIMENSION NX*NY, NX*NZ, OR NY*NZ DICTATED BY IPLANE; &
C      NDIM SETS VALUE OF DIMENSION OF GARRAY.
C      *USE FLD2DA(TITLE,GARRAY,NY,NX) TO PRINT ANY ARRAY DIMENSIONED
C      GARRAY(NY,NX); SET 'TITLE' TO REQUIRED NAME ( 4 HOLLERITH
C      CHARACTERS ONLY).
C      *USE FLD3DA(TITLE,GARRAY,NX,NY,NZ,IPLANE,ILOC) TO PRINT ANY
C      ARRAY DIMENSIONED GARRAY(NX,NY,NZ) IN PLANE SPECIFIED BY
C      'IPLANE' & 'ILOC' AS FOR CONTUR ABOVE; SET 'TITLE' AS FOR
C      FLD2DA.
C      VARIABLE NAMES FOR USE IN GROUND:
C      COMMON/TYPE/CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME,WALL
C      COMMON/VAR/P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
C      &KE,EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
C      COMMON/VAROLD/P10,PP0,U10,U20,V10,V20,W10,W20,R10,R20,RS0,
C      &KE0,EPO,H10,H20,H30,C10,C20,C30,C40,RX0,RY0,RZ0,S10,S20
C      COMMON/VARLOW/P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
C      &KEL,EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
C      COMMON/VARHI/P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
C      &KEH,EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
C      COMMON/GMTRY/VOL,VOLO,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
C      COMMON/PROP/D1,D2,D1DP,D2DP,MU1,MU1LAM,EXCO,CFP,MDT,HST1,HST2
C      COMMON/PRPOLD/D10,D20
C      COMMON/PRPLOW/D1L,D2L,EXCOL
C      COMMON/PRPHI/D1H,D2H,MU1H,EXCOH
C      COMMON/VARNX/XG,XU,DXU,DXG
C      COMMON/VARNY/YG,YV,DYV,DYG,R,RV
C      COMMON/VARNZ/ZG,ZW1,DZW,DZG,WGRID
C      COMMON/GDMSCI/XPLANE,YPLANE,ZPLANE,ITNO
C      COMMON/GDMSCL/LSLAB,MSLAB,HSLAB,LAMMU
C      REAL NORTH,LOW
C      INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
C      &EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
C      INTEGER P10,PP0,U10,U20,V10,V20,W10,W20,R10,R20,RS0,
C      &EPO,H10,H20,H30,C10,C20,C30,C40,RX0,RY0,RZ0,S10,S20
C      INTEGER P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
C      &EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
C      INTEGER P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
C      &EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
C      INTEGER VOL,VOLO,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
C      INTEGER D1,D1DP,D2,D2DP,EXCO,CFP,HST1,HST2
C      INTEGER D10,D20,D1L,D2L,EXCOL,D1H,D2H,EXCOH
C      INTEGER XG,XU,DXU,DXG,YG,YV,DYV,DYG,R,RV,ZG,ZW1,DZW,
C      &DZG,WGRID
C      INTEGER XPLANE,YPLANE,ZPLANE
C      LOGICAL LSLAB,MSLAB,HSLAB,LAMMU,LSPDA
C      EQUIVALENCE (M1,R1),(M2,R2)
C      SATLIT-EQUIVALENT IRUN:
C      EQUIVALENCE (IRUN,INTGR(11))
C      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
C      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C      ARRAYS ( DIMENSIONED NY,NX ) FOR USE WITH 'ADD':
C      DIMENSION CVAR(1,1),VVAR(1,1),CM(1,1),VM(1,1),ZERO(1,1)
C      DIMENSION GP(30,1),GH(30,1),GD(30,1),GV(30,1),GW(30,1)

```



```

1 ,GMACH(30,1),GTEMP(30,1),GVISC(30,1),GWH(30,1),GWM(30,1)
2 ,GKE(30,1),GC3(30,1),GYG(30,1),GXX(30,1),GYY(30,1),GZZ(30,1)
C SPECIAL-DATA ARRAYS DIMENSIONED & DIMENSION VALUES SET HERE:
C DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
C USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
EQUIVALENCE (RAIR,RE(21)),(GAMA,RE(22)),(GSWP,RE(23)),
1(GPR,RE(24)),(GTW,RE(25)),(GEMU1,RE(26)),(JEMU1,INTGR(1))
DATA NLSP,NISP,NRSP/1,1,1/
DATA CVAR,VVAR,CM,VM,ZERO/5*0.0/
C USER PLACES HIS DATA STATEMENTS HERE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
C PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C STATEMENTS OF THIS SECTION.
DATA NUMCH4 / 0 /
IF(SPDATA)
&CALL RDSPC(IRN,INTGR(12),LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
CALL GRDUTY(IRN,ICHAP,IZED,INDVAR)
IF(BFC) CALL BFCGRD(IRN,ICHAP,ISWP,IZED,INDVAR)
IF(ICHAP.EQ.-5) GO TO 10
IF(ICHAP.LE.0.OR.ICHAP.GT.16) RETURN
GO TO (100,200,300,4999,500,600,700,800,900,1000,1100,1200,
&1300,1400,1500,1600),ICHAP
RETURN
4999 NUMCH4= NUMCH4 + 1
IF (MOD(NUMCH4,2).EQ.1) GO TO 400
RETURN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C-----
C CHAPTER 0: MODIFY SATLIT DATA, AT START OF EACH IRN.
C-----
10 CONTINUE
IF(.NOT.NAMLST) RETURN
IF(IRN.EQ.NRUN) DATFIL=.FALSE.
C--- READ SATLIT DATA NAMELIST HERE
CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 1 TO 24 )
C READ(20,G1G24)
CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 25 TO 42 )
C READ(20,G25G42)
C RETURN
C-----
C CHAPTER 1: CALLED AT THE START OF EACH TIME STEP.
C SET 'DT' HERE WHEN TLAST SET NEGATIVE IN BLOCK DATA.
C 'ATIME + DT' GIVES THE END TIME OF THE CURRENT TIME STEP.
C NOT ACCESSED IF STEADY,OR PARABOLIC.
C-----
100 CONTINUE
RETURN
C-----
C CHAPTER 2: CALLED AT THE START OF EACH SWEEP.
C-----
200 CONTINUE
RETURN
C-----
C CHAPTER 3: CALLED AT THE START OF EACH SLAB;
C NOT ACCESSED IF PARABOLIC, BUT 'STRIDE' IS.
C-----
300 CONTINUE
RETURN
C-----
C CHAPTER 4: CALLED AT THE START OF EACH RE-CALCULATION OF
C VARIABLES P1,...C4 AT CURRENT SLAB. ITNO= ITERATION NUMBER.
C-----
400 CONTINUE
RETURN
C-----
C CHAPTER 5: GROUND CALLED WHEN SOURCE TERM IS COMPUTED.
C INDVAR GIVES DEPENDENT VARIABLE IN QUESTION IE. U1,...C4.
C TO ADD SOURCE TO DEPENDENT VARIABLE C1(SAY) FOR IX=IXF,IXL
C AND IY=IYF,IYL INSERT STATEMENT:
C IF(INDVAR.EQ.C1)

```

VANO1450
VANO1460
VANO1470
VANO1480
VANO1490
VANO1500
VANO1510
VANO1520
VANO1530
VANO1540
VANO1550
VANO1560
VANO1570
VANO1580
VANO1590
VANO1600
VANO1610
VANO1620
VANO1630
VANO1640
VANO1650
VANO1660
VANO1670
VANO1680
VANO1690
VANO1700
VANO1710
VANO1720
VANO1730
VANO1740
VANO1750
VANO1760
VANO1770
VANO1780
VANO1790
VANO1800
VANO1810
VANO1820
VANO1830
VANO1840
VANO1850
VANO1860
VANO1870
VANO1880
VANO1890
VANO1900
VANO1910
VANO1920
VANO1930
VANO1940
VANO1950
VANO1960
VANO1970
VANO1980
VANO1990
VANO2000
VANO2010
VANO2020
VANO2030
VANO2040
VANO2050
VANO2060
VANO2070
VANO2080
VANO2090
VANO2100
VANO2110
VANO2120
VANO2130
VANO2140
VANO2150
VANO2160

```

C      &CALL ADD(INDVAR,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C      NOTES ON 'ADD':
C      *SOURCE= (CVAR(IY,IX)+AMAX1(0.0,MASFLO))*(VVAR(IY,IX)-PHI),
C      WHERE 'PHI' IS IN-CELL VALUE OF VARIABLE IN QUESTION.
C      *'MASFLO'= CM(IY,IX)*(VM(IY,IX)-P),
C      WHERE 'P' IS THE IN-CELL PRESSURE.
C      *FOR INDVAR= M1, OR =M2, SOURCE ADDED IS 'MASFLO' ONLY,
C      EXCEPT FOR ONEPHS=.F. & MASFLO < 0.0 (IE. OUTFLOW) WHEN
C      CM(IY,IX) IS MULTIPLIED BY R1*D1 (FOR M1) & R2*D2 (FOR M2).
C      *BOTH 'CVAR' & 'CM' ARE MULTIPLIED BY CELL-GEOMETRY QUANTITY
C      DICTATED BY SETTING OF 'TYPE' (=CELL, EAST AREA,..VOLUME).
C      *TYPE-SPECIFIED AREAS ARE CALCULATED AS IF BLOCKAGE ABSENT,
C      BUT 'VOLUME' WITH ACCOUNT FOR ITS PRESENCE.
C      *FOR ALL SOLVED VARIABLES, INCLUDE DING M1 ( & M2 WHEN ONEPHS=F),
C      IF 'CM' > 0.0 CALL 'ADD'; FOR M1 & M2 ALTHOUGH 'CVAR' & 'VVAR'
C      HAVE NO SIGNIFICANCE THEY MUST BE ENTERED AS ARGUMENTS.
C      *'CVAR', 'VVAR', 'CM' & 'VM' MUST BE DIMENSIONED NY,NX.
C-----
500 CONTINUE
   RETURN
C-----
C      CHAPTER 6: CALLED AT THE END OF EACH VARIABLE-RECALCULATION
C      CYCLE COMMENCED AT CHAPTER 4. ITNO = ITERATION NUMBER.
C-----
600 CONTINUE
   RETURN
C-----
C      CHAPTER 7: CALLED AT END OF EACH SLAB-WISE CALCULATION.
C-----
700 CONTINUE
  IF(FLOAT(ISWP).LT.GSWP) RETURN
  CALL GET(P1,GP,NY,NX)
  CALL GET(H1,GH,NY,NX)
  CALL GET(D1,GD,NY,NX)
  CALL GET(V1,GV,NY,NX)
  CALL GET(W1,GW,NY,NX)
  CALL GET(KE,GKE,NY,NX)
C  CALL GETID(YG,GYG,NY)
  CALL GRED1(39,IZED,GYG,NY,NX)
  CALL GRED3(57,IZED,GXX,GYY,GZZ,NY,NX)
  GCP=RAIR/(1.-1/GAMA)
  DO 701 I=1,NY
    GSON=SQRT(GAMA*GP(I,1)/GD(I,1))
    GAV=SQRT(GV(I,1)**2+GW(I,1)**2)
    GMACH(I,1)=GAV/GSON
  701 GTEMP(I,1)=GP(I,1)/GD(I,1)/RAIR
C 701 GTEMP(I,1)=(GH(I,1)-GW(I,1)**2/2.-GV(I,1)**2/2.)/GCP
    CALL SET(C1,1,NX,1,NY,GMACH,NY,NX)
    CALL SET(C2,1,NX,1,NY,GTEMP,NY,NX)
C-----CALCULATE DY1 CF ST H(CONVECTIVE COEF.) Q TAU TR
  IF(JEMUL.NE.2) GOTO 702
C-----TURBULENT VALUES
  GCF=2./GW(NY,1)**2*GKE(1,1)/3.33*GD(1,1)/GD(NY,1)
C7  GCF=GCF*GD(NY,1)/GD(1,1)*GTEMP(NY,1)/GTEMP(1,1)*GP(1,1)/GP(NY,1)
  GST=GCF/2./GPR**.666
  GHH=GD(NY,1)*GCP*GW(NY,1)*GST
  GR=GPR**3.33
  GTR=GTEMP(NY,1)*(1.+GR*(GAMA-1.)/2.*GMACH(NY,1)**2)
C  1(1.+(GAMA-1.)/2.*GMACH(NY,1)**2)
  GQ=GHH*(GTR-GTW)
  GOTO 703
C-----LAMINAR VALUES
702 CONTINUE
  IF(JEMUL.EQ.-1) GEMUL=GVIS(1,1)
  GQ=GEMUL/GPR*(GH(1,1)-GTW*GCP)/GYG(1,1)
  GR=GPR**5
  GTR=GTEMP(NY,1)*(1.+GR*(GAMA-1.)/2.*GMACH(NY,1)**2)
C  1(1.+(GAMA-1.)/2.*GMACH(NY,1)**2)
  GHH=GQ/(GTR-GTW)
  GST=GHH/(GD(NY,1)*GW(NY,1)*GCP)
  GTAU=GEMUL*GW(1,1)/GYG(1,1)
  GCF=GTAU*2./(GD(NY,1)*GW(NY,1)**2)

```

VAN02170
 VAN02180
 VAN02190
 VAN02200
 VAN02210
 VAN02220
 VAN02230
 VAN02240
 VAN02250
 VAN02260
 VAN02270
 VAN02280
 VAN02290
 VAN02300
 VAN02310
 VAN02320
 VAN02330
 VAN02340
 VAN02350
 VAN02360
 VAN02370
 VAN02380
 VAN02390
 VAN02400
 VAN02410
 VAN02420
 VAN02430
 VAN02440
 VAN02450
 VAN02460
 VAN02470
 VAN02480
 VAN02490
 VAN02500
 VAN02510
 VAN02520
 VAN02530
 VAN02540
 VAN02550
 VAN02560
 VAN02570
 VAN02580
 VAN02590
 VAN02600
 VAN02610
 VAN02620
 VAN02630
 VAN02640
 VAN02650
 VAN02660
 VAN02670
 VAN02680
 VAN02690
 VAN02700
 VAN02710
 VAN02720
 VAN02730
 VAN02740
 VAN02750
 VAN02760
 VAN02770
 VAN02780
 VAN02790
 VAN02800
 VAN02810
 VAN02820
 VAN02830
 VAN02840
 VAN02850
 VAN02860
 VAN02870
 VAN02880

703	GC3(1,1)=GYG(1,1)	VAN02890
	GC3(2,1)=GCF	VAN02900
	GC3(3,1)=GST	VAN02910
	GC3(4,1)=GCF/2./GST	VAN02920
	GC3(5,1)=GHH	VAN02930
	GC3(6,1)=GQ	VAN02940
	GC3(7,1)=GTAU	VAN02950
	GC3(8,1)=GTR	VAN02960
	GC3(9,1)=GTR-GTW	VAN02970
	GC3(10,1)=GD(NY,1)*GW(NY,1)*GZZ(1,1)/GEMU1	VAN02980
	GC3(11,1)=GZZ(1,1)	VAN02990
	GC3(12,1)=GEMU1	VAN03000
	GC3(13,1)=GD(NY,1)*GW(NY,1)*GYG(1,1)/GEMU1*SQRT(ABS(GCF/2.))	VAN03010
	CALL SET(C3,1,NX,1,NY,GC3,NY,NX)	VAN03020
	RETURN	VAN03030
C	-----	VAN03040
C	CHAPTER 8: CALLED AT THE END OF EACH SWEEP;	VAN03050
C	NOT ACCESSED IF PARABOLIC.	VAN03060
C	-----	VAN03070
800	CONTINUE	VAN03080
	RETURN	VAN03090
C	-----	VAN03100
C	CHAPTER 9: CALLED AT THE END OF EACH TIME STEP;	VAN03110
C	NOT ACCESSED IF PARABOLIC.	VAN03120
C	-----	VAN03130
900	CONTINUE	VAN03140
	RETURN	VAN03150
C	-----	VAN03160
C	CHAPTER 10: SET PHASE 1 DENSITY HERE WHEN IRH01=-1 IN DATA.	VAN03170
C	SET CURRENT-Z 'SLAB' DENSITY, D1, IF MSLAB=.T.,	VAN03180
C	EG. IF(MSLAB) CALL SET(D1,1,NX,1,NY,GD1,NY,NX).	VAN03190
C	SET NEXT LARGER-Z 'SLAB' DENSITY, D1H, IF HSLAB=.T. & PARAB=F	VAN03200
C	EG. IF(HSLAB) CALL SET(D1H,1,NX,1,NY,GD1H,NY,NX).	VAN03210
C	SET D(LN(D1))/DP (IE. D1DP) FOR UNSTEADY FLOW,	VAN03220
C	EG. IF(MSLAB) CALL SET(D1DP,1,NX,1,NY,GD1DP,NY,NX).	VAN03230
C	-----	VAN03240
1000	CONTINUE	VAN03250
	IF (MSLAB) GO TO 101	VAN03260
	JP1=PIH	VAN03270
	JH1=H1H	VAN03280
	JD1=D1H	VAN03290
	JW1=W1H	VAN03300
	JV1=V1H	VAN03310
	GO TO 102	VAN03320
101	JP1=P1	VAN03330
	JH1=H1	VAN03340
	JD1=D1	VAN03350
	JW1=W1	VAN03360
	JV1=V1	VAN03370
102	CALL GET(JP1,GP,NY,NX)	VAN03380
	CALL GET(JH1,GH,NY,NX)	VAN03390
	CALL GET(JW1,GW,NY,NX)	VAN03400
	CALL GET(JV1,GV,NY,NX)	VAN03410
	IF(IZED.EQ.1) GOTO 105	VAN03420
	IF(IZED.EQ.NZ) GOTO 109	VAN03430
C	----- IZED=2, NZ=1	VAN03440
	DO 103 IX=1,NX	VAN03450
	DO 103 IY=1,NY	VAN03460
	IF(HSLAB) GOTO 104	VAN03470
	GWA=(GW(IY,IX)+GWM(IY,IX))/2.	VAN03480
	GWM(IY,IX)=GW(IY,IX)	VAN03490
	GOTO 115	VAN03500
104	GWA=(GW(IY,IX)+GWH(IY,IX))/2.	VAN03510
	GWH(IY,IX)=GW(IY,IX)	VAN03520
115	GHS=GH(IY,IX)-(GWA**2+GV(IY,IX)**2)/2.	VAN03530
	IF(GHS.LE.1.E5) GHS=1.E5	VAN03540
103	GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS	VAN03550
	GOTO 113	VAN03560
C	----- IZED=1	VAN03570
105	DO 106 IX=1,NX	VAN03580
	DO 106 IY=1,NY	VAN03590
	GHS=GH(IY,IX)-(GW(IY,IX)**2+GV(IY,IX)**2)/2.	VAN03600

```

      IF(GHS.LE.1.E5) GHS=1.E5
      GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS
      IF(HSLAB) GOTO 107
      GWM(IY,IX)=GW(IY,IX)
      GOTO 106
107  GWH(IY,IX)=GW(IY,IX)
106  CONTINUE
      GOTO 113
C-----I ZED=NZ
109  DO 110 IX=1,NX
      DO 110 IY=1,NY
      IF(HSLAB) GOTO 111
      GHS=GH(IY,IX)-(GWM(IY,IX)**2+GV(IY,IX)**2)/2.
      IF(GHS.LE.1.E5) GHS=1.E5
      GWM(IY,IX)=GW(IY,IX)
      GOTO 112
111  GHS=GH(IY,IX)-(GWH(IY,IX)**2+GV(IY,IX)**2)/2.
C    IF(GHS.LE.1.E5) GHS=1.E5
      GWH(IY,IX)=GW(IY,IX)
112  GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS
110  CONTINUE
C-----
113  CONTINUE
      CALL SET(JD1,1,NX,1,NY,GD,NY,NX)
      RETURN
C-----
C    CHAPTER 11: SET PHASE 2 DENSITY HERE WHEN IRHO2=-1 IN DATA.
C    SET CURRENT-Z 'SLAB' DENSITY, D2, IF MSLAB=.T.,
C    EG. IF(MSLAB) CALL SET(D2,1,NX,1,NY,GD2,NY,NX).
C    SET NEXT LARGER-Z 'SLAB' DENSITY, D2H, IF HSLAB=.T. & PARAB=F
C    EG. IF(HSLAB) CALL SET(D2H,1,NX,1,NY,GD2H,NY,NX).
C    SET D(LN(D2))/DP FOR UNSTEADY FLOW,
C    EG. IF(MSLAB) CALL SET(D2DP,1,NX,1,NY,GD2DP,NY,NX).
C-----
1100 CONTINUE
      RETURN
C-----
C    CHAPTER 12: SET PHASE 1 VISCOSITY HERE WHEN IEMU1=-1 IN DATA.
C    SET CURRENT-Z 'SLAB' VISCOSITY (MU1), IF MSLAB=.T.,
C    EG. IF(MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX).
C    SET NEXT LARGER-Z 'SLAB' VISC. (MU1H), IF HSLAB=.T. & PARAB=F
C    EG. IF(HSLAB) CALL SET(MU1H,1,NX,1,NY,GVISC,NY,NX).
C-----
C    CHAPTER ALSO ACCESSED WHEN EMULAM=-1.0 IN DATA, SO THAT THE
C    LAMINAR VISCOSITY WHICH APPEARS IN WALL FUNCTIONS & IN THE
C    KE-EP TURBULENCE MODEL (IEMU1=2) MAY BE SET NON-CONSTANT.
C    SET CURRENT-Z 'SLAB' VALUE (MU1LAM) WHEN LAMMU=.T.,
C    EG. IF(LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GVISC,NY,NX).
C-----
1200 CONTINUE
      GCP=RAIR/(1.-1/GAMA)
      IF (HSLAB) GOTO 122
      CALL GET(H1,GH,NY,NX)
      CALL GET(W1,GW,NY,NX)
      CALL GET(V1,GV,NY,NX)
      GOTO 123
122  CALL GET(H1H,GH,NY,NX)
      CALL GET(W1H,GW,NY,NX)
      CALL GET(V1H,GV,NY,NX)
123  CONTINUE
      DO 121 IX=1,NX
      DO 121 IY=1,NY
      GTMP=(GH(IY,IX)-GW(IY,IX)**2/2.-GV(IY,IX)**2/2.)/GCP
      IF(GTMP.LT.150.) GTMP=150.
121  GVISC(IY,IX)=1.716E-05*(GTMP/273.)*.666
C121 IF(GVISC(IY,IX).LE..8E-5) GVISC(IY,IX)=.8E-5
      IF (MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX)
      IF (HSLAB) CALL SET(MU1H,1,NX,1,NY,GVISC,NY,NX)
      IF (LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GVISC,NY,NX)
      RETURN
C-----
C    CHAPTER 13: SET EXCHANGE COEFFICIENT (E.C.) FOR VARIABLE

```

VAN03610
 VAN03620
 VAN03630
 VAN03640
 VAN03650
 VAN03660
 VAN03670
 VAN03680
 VAN03690
 VAN03700
 VAN03710
 VAN03720
 VAN03730
 VAN03740
 VAN03750
 VAN03760
 VAN03770
 VAN03780
 VAN03790
 VAN03800
 VAN03810
 VAN03820
 VAN03830
 VAN03840
 VAN03850
 VAN03860
 VAN03870
 VAN03880
 VAN03890
 VAN03900
 VAN03910
 VAN03920
 VAN03930
 VAN03940
 VAN03950
 VAN03960
 VAN03970
 VAN03980
 VAN03990
 VAN04000
 VAN04010
 VAN04020
 VAN04030
 VAN04040
 VAN04050
 VAN04060
 VAN04070
 VAN04080
 VAN04090
 VAN04100
 VAN04110
 VAN04120
 VAN04130
 VAN04140
 VAN04150
 VAN04160
 VAN04170
 VAN04180
 VAN04190
 VAN04200
 VAN04210
 VAN04220
 VAN04230
 VAN04240
 VAN04250
 VAN04260
 VAN04270
 VAN04280
 VAN04290
 VAN04300
 VAN04310
 VAN04320

C	INDVAR WHEN SIGMA(INDVAR)=-1.0 IN DATA.	VAN04330
C	SET CURRENT-Z 'SLAB' E.C. (EXCO) IF MSLAB=.T.,	VAN04340
C	EG. IF(MSLAB) CALL SET(EXCO,1,NX,1,NY,GEXCO,NY,NX).	VAN04350
C	SET NEXT SMALLER-Z 'SLAB' E.C. (EXCOL) IF LSLAB=.T.,	VAN04360
C	EG. IF(LSLAB) CALL SET(EXCOL,1,NX,1,NY,GEXCOL,NY,NX).	VAN04370
C	SET NEXT LARGER-Z 'SLAB' E.C. (EXCOH) IF HSLAB=.T.,	VAN04380
C	EG. IF(HSLAB) CALL SET(EXCOH,1,NX,1,NY,GEXCOH,NY,NX).	VAN04390
C	NOTE: FOR MSLAB, INDVAR=U1,..C4; FOR LSLAB, INDVAR=U1L,..C4L	VAN04400
C	& FOR HSLAB, INDVAR=U1H,..C4H. IF PARAB=.T. SET MSLAB ONLY.	VAN04410
C	-----	VAN04420
	1300 CONTINUE	VAN04430
	RETURN	VAN04440
C	-----	VAN04450
C	CHAPTER 14: SET INTER-PHASE FRICTION COEFFICIENT (CFP) HERE	VAN04460
C	WHEN ICFIP = -1 IN DATA; ITS UNITS = FORCE / (CELL * RELATIVE	VAN04470
C	SPEED OF PHASES).	VAN04480
C	-----	VAN04490
	1400 CONTINUE	VAN04500
	RETURN	VAN04510
C	-----	VAN04520
C	CHAPTER 15: SET INTER-PHASE MASS-TRANSFER RATE PER CELL (MDT)	VAN04530
C	HERE WHEN IMDOT = -1 IN DATA.	VAN04540
C	-----	VAN04550
	1500 CONTINUE	VAN04560
	RETURN	VAN04570
C	-----	VAN04580
C	CHAPTER 16: SET HERE PHASE 1 & 2 SATURATION ENTHALPIES	VAN04590
C	(HST1 & HST2) WHEN IHSAT = -1 IN DATA.	VAN04600
C	-----	VAN04610
	1600 CONTINUE	VAN04620
	RETURN	VAN04630
	END	VAN04640

INITIAL DISTRIBUTION LIST

	No. Copies
1. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
2. Research Administration Code 012 Naval Postgraduate School Monterey, CA 93943	1
3. Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	1
4. Professor Matthew D. Kelleher, Code 69Kk Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	2
5. Professor Robert H. Nunn, Code 69Nn Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	2
6. Professor David Salinas, Code 69Sa Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	1
7. A. Leitner Visiting Research Associate Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	3
8. Dr. R. Dillinger Naval Weapon Center China Lake, CA 93555	1
9. Dr. C. Porter Naval Weapon Center China Lake, CA 93555	1
10. Mr. A. Danielson Naval Weapon Center China Lake, CA 93555	1

U225531

DUDLEY KNOX LIBRARY - RESEARCH REPORTS



5 6853 01071179 9

~~U225531~~